



**United International University**  
Department of Computer Science and Engineering  
CSI 221 Database Management Systems, Mid Exam, Fall 2019  
Total Marks: 30, Time: 1 hour 45 minutes

**Answer all the questions**

1. a) Draw an *E/R Diagram* for the following business scenario of a *Book Reviewing Website*: [7]

- The system stores all the books title, author, ISBN, category, publisher, publish\_date, language and awards. Each book is identified by its ISBN no and may have more than one awards.
- Each book reader of the system is identified by his email address. The system also stores each reader's encoded password, join\_date, gender and address. Readers can track books with *reading or, have\_read or, want\_to\_read* statuses. The system also recommend books for each reader. Each reader can follow other book readers too and each reader can also be followed by more than one reader.
- Each reader can post review of a book. The review contains a unique review id, datetime and details. Each review is associated with a single book.

b) Now create an equivalent Relational Schema for the E/R Diagram drawn in Q(1.a). [4]

2. Show the results of each query for the given tables: [2.5+2.5]

**Player:**

id	salary	club_id	rank
2	20000	5	1
3	6000	3	3
1	15000	6	2
5	9000	5	3

**Club:**

club_id	country_name	coach_id	budget
1	Bangladesh	2	30000
5	Brazil	3	20000
10	France	6	20000
6	Brazil	10	50000

a)

```
SELECT C.club_id,  
       MAX(P.salary) AS max_sal  
FROM   Club C  
       JOIN  
       Player P  
       ON C.club_id=P.club_id  
WHERE P.salary>11000  
GROUP BY C.club_id  
ORDER BY max_sal;
```

b)

```
SELECT *  
FROM Club AS c1  
WHERE 2 = (  
    SELECT COUNT(DISTINCT budget)  
    FROM Club AS c2  
    WHERE c2.budget>c1.budget  
);
```

3. Consider the following Relational Database Schema:

Table Name	Columns						
riders	<u>id</u> ,	name,	join_date,	vehicle_no,	rating		
passengers	<u>id</u> ,	name,	dob,	address			
rides	<u>rider_id</u> (FK),	<u>passenger_id</u> (FK),	<u>datetime</u> ,	fare,	discount,	src,	dest

Now solve the following user queries using MySQL:

- a) Answer any 2: [1.5+1.5]
- Show the name and join\_date of those riders whose name consists of minimum 4 characters and whose joining year is 2019. [If necessary use YEAR() function]
  - Show the source, destination and final fare of those rides where the final fare is greater than 5000/= [ Here, final fare= fare\*(1-discount/100) ]
  - Show the id, name and vehicle number of those riders who have rating values between 3 and 5.
- b) Answer any 1: [2]
- Show the name of those riders who have made at least one ride at destination 'Dhaka'. Your query should output each rider's name only one time.
  - Show the datetime, passenger name and rider name of each ride. Don't consider those rides where the rider has rating less than 2 or, the passenger's date of birth is on or after 2000-01-01.
- c) Answer any 1: [2]
- For each year, show the total number of riders joined during that year. Exclude those riders from your count having rating value less than 3. Show the year number and total counts of such riders.
  - For each source and destination, show the total number of rides were taken during the year 2019. Show source, destination and no of rides data only. Avoid those source and destinations having less than 20 rides during the year 2019.
- d) Answer any 1: [2]
- For each vehicle, show the vehicle no and the total no of rides conducted by that vehicle for only those vehicles whose name don't start with 'GA' and participated in at least 3 rides.
  - Show the maximum number of rides conducted by any rider.
- e) Answer any 2: [2.5+2.5]
- For each year, show the rider id(s) riding maximum number of rides. [if necessary, use YEAR function]
  - Show the passenger id(s) who has spent maximum total amount of fare in his rides.
  - Show those rider id(s) whose total income is greater than the average income of all the other riders. [here, total income = total fare amount]