Set	**:**	**B**
Unique Paper Code	**:**	**32341403**
Course	**:**	**B. Sc. (H) Computer Science**
Paper	**:**	**Database Management Systems**
Semester	**:**	**IV**
Duration	**:**	**2 Hours**
Maximum Marks	**:**	**75**

Attempt any FOUR questions

All questions carry equal marks

(For courses effective from Academic Year 2015-16)

Q1 Consider the following case study:

`BookMyEvent` is an online booking website which allows customers to book tickets for events, shows and artists' performances.

The site stores details about various performance artists that the customers might be interested in such as the artist's name, gender, date of birth, latest work. Artists are associated to an artist category, namely, comedy, tragedy, theatre, T.V. artist. Each category has a code, a title and a description and most categories have several artists, although some are not populated yet.

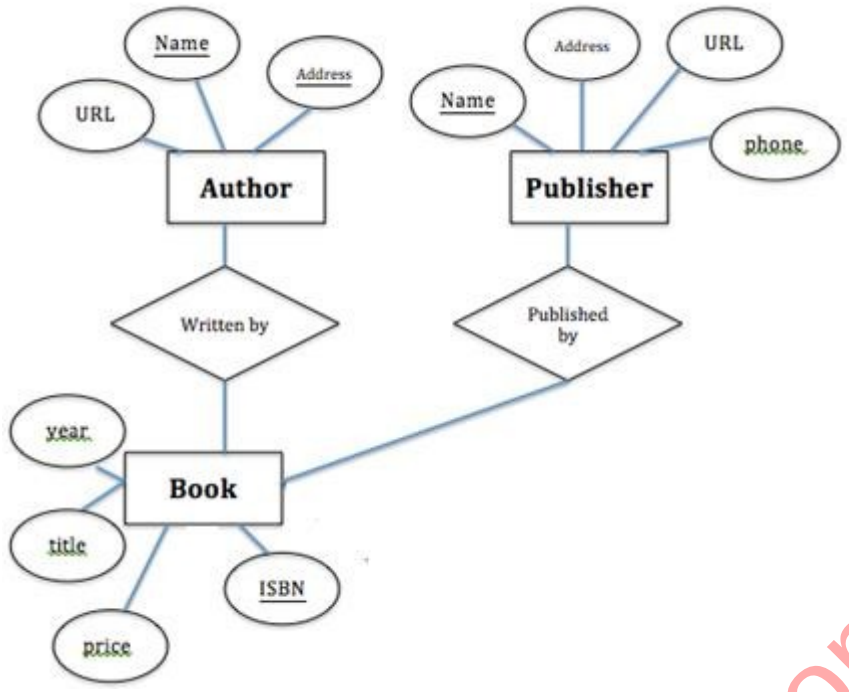
Artists may perform at many different events. All events have an id, a name, a description, start time, and duration. An event may have many news stories released about it, namely, 'new show in city' or 'successfully running for many weeks', to develop interest in the events. Event news has an id, title, author and date.

Events are held at a specific venue. Venues have a name, address, lead contact, seating capacity, star rating. Star ratings are stored with a number, **`1-5`**.

Many tickets are sold for each event. Tickets include a number, type, such as standard or VIP and a price. Each ticket is for one event only. Customers have to register on the system before they can reserve tickets. They must add standard information including name, address, email, and contact number. A customer can buy many tickets for an event though a ticket is for one event and one customer.

For the above given case study, identify the entities and relationships with their associated attributes, cardinality ratio, and participation constraints. Also, identify subclasses for entities, if they exist. Then, based on these, draw an Enhanced Entity-Relationship Diagram. Clearly state any assumptions you make.

Q2



Study the above ER diagram. For each of the relationships, find the cardinality ratio, and participation constraint values. Map the ER diagram to the relation schema. Clearly show all the steps. Mention all the primary and foreign keys for each relation. Clearly state any assumptions that are made.

Q3 You are hired as a database designer for an **animal adoption agency**. You designed the following relations.

```
Animal (AnimalID, Name, DateAdmitted, PreviousOwnerID)
Adopter (ID, Name, Address, OtherAnimals)
Adoption (AnimalID, AdoptDate, PrevOwnerID,
CurrentAdopterID)
MedicalRecord (AnimalID, age, vaccine, prescription,
dosage)
```

- The underlined attributes are the primary keys.
- The relation **Animal** stores the information about the animal. **DateAdmitted** is of **Date** data type. The **PreviousOwnerID** value should be **NULL** by default. It references the **ID** attribute in the relation **Adopter**.
- The **Adopter** relation stores information about the person who adopts the animal. The attribute **OtherAnimals** should allow values as **Yes** or **No** only.
- The **Adoption** relation depicts the relationship between the relations **Animal** and **Adopter**. **AdoptDate** should be of **Date** data type and should not be **NULL**. **PrevOwnerID** and **CurrentAdopterID** refer to the **ID** attribute of the relation **Adopter** and may or may not be the same.
- **AnimalID** attribute in the **MedicalRecord** relation references the **AnimalID** attribute in the **Animal** relation.

For the above relations, answer the following questions using SQL:

- Write **CREATE TABLE** command for each of the tables. You must use data types, **PRIMARY KEY** constraint, and **FOREIGN KEY** constraint, as applicable. Implement the other constraints mentioned above
- For each of the given relations, write a command to **INSERT** one row with appropriate values

Q4 Consider the relations in question **3**.

Design **four** queries in English language. The **first** query should involve a join and should display the results in a sorted fashion. The **second** query should involve an aggregation operator and the **group by** clause. The **third** query should update the data (based on a condition) in one of the tables. The **fourth** query should delete all the data in one of the tables.

Write the corresponding SQL queries for **any of the three** English queries.
Write relational algebra queries for **any one of the** English queries.

Example: If you were to write a query using **WHERE** clause, you might answer as follows:
English query: ***Find name of the animal whose AnimalID is 234567.***
SQL query:
```
SELECT Name
FROM Animal
WHERE AnimalID = 234567;
```

Q5 Consider a relation **R (A, B, C, D, E, F)** with the following set of functional dependencies

$$F = \{AB \rightarrow C,\ DC \rightarrow AE,\ E \rightarrow F\}$$

Find any two keys for the given relation? Is this relation in **2NF**? If not, state the reason and decompose this relation until each of the decomposed relations are in **2NF**.

Q6 Consider a disk with block size **B = 1024 bytes**. A block pointer is **P = 8 bytes** long and a record pointer is P_R **= 10 bytes** long. A file has **r = 102,000 Book** records of fixed length. Each record has the following fields: **BookID (6 bytes), Name (30 bytes), Author (30 bytes), Publisher (30 bytes), Year (2 bytes), PrintedCopies (2 bytes).** Assume that the file is ordered by the key field **BookID**.

- Calculate the record size **R** for the file in bytes.
- Assuming the un-spanned organization, calculate the blocking factor and the number of file blocks **b** required to store all the records.
- Calculate the number of block accesses required for searching a record in the data file using binary search.

Suppose that we construct a primary index on **BookID** for the above file.

- Calculate the size of an entry in the index R_i.
- Calculate the index blocking factor bfr_i
- Find the total number of index entries r_i and the number of index blocks, **bi**.
- Calculate the number of block accesses required for searching a record in the data file using this primary index.