National University of Computer and Emerging Sciences

Islamabad Campus

Part A – SQL, Relational Algebra, Mapping
Attempt all the questions.

Q1: **[10 x 2 = 20 marks]**

You are working on a database containing tables for:

Book (Book Id, ISBN, Title, Publisher, Year)
Student (Student Id, Student_Name, Course_Name, Age)
Author (Author Name, Age, Email)
Borrow (Book Id*, Student Id*, Borrowed on)
Wrote (Book Id*, Author Name*)
Classification (Book Id*, Genre_Name*)
Genre (Genre_Name, Description)

The key fields are underlined, and the field name with * represents foreign keys. Write the following queries in relational algebra from the above database.

a) **List all authors, by name, who are aged over 40.**

ÕAuthor Name (sAge>40(Author))

b) **List the names of all authors who wrote books in the "Science Fiction" genre borrowed in October 2021.** ***Note: Any sensible date format can be used***

ÕAuthor Name, Email (

sGenre Name = "Science Fiction" ^ Borrowed on >= "01/10/2021" ^ Borrowed on <= "31/10/2021"

(Author ⋈Author_Name

(Wrote ⋈Book_ID

(Borrow ⋈Book_ID

(Book ⋈Book_ID Classification)))))

c) **List all student names and their course name for all students who borrowed a book between 3rd December 2021 and 17th December 2021 inclusively.** ***Note: Any sensible date format can be used***

ÕStudentName,CourseName(

s(Borrowed_on >= "03/12/2021" ^ Borrowed_on <= "17/12/2021") (Student ⋈StudentID Borrow))

d) List the authors (by name), who have written a book borrowed by Student Peter Sage. We only require books that were published in 2012 and that are categorised as Poetry.

ÕAuthor_Name(

s(Student_Name=”Peter Sage” ^ Year = 2012 ^ Genre_Name = “Poetry”)

(Student ⋈Student_ID

(Borrow ⋈Book_ID

(Wrote ⋈Book_ID

(Book ⋈ Book_ID Classification)))))

e) List all genres and descriptions of books written by authors between the ages of 30 and 40 inclusively.

ÕGenre,Description (

sAge≥30 ^ Age ≤40

(Author ⋈Author_Name

(Wrote ⋈Book_ID

(Book⋈Book_ID

(Classification⋈Genre_NameGenre)))))

f) Determine all students who have borrowed books but are not enrolled in any course.

πStudent.Student_Name((Student⋈Borrow)−(Student⋈Course))

g) List the titles of books borrowed by students enrolled in courses related to 'Engineering' or 'Computer Science':

πBook.Title((Student⋈Borrow⋈Book)⋈(σStudent.Course_Name=’Engineering’∨Student.Course_Name=’Computer Science’(Student)))

h) Find all books that have not been borrowed by any student under the age of 20.

πBook.Title(Book−πBook.Book_Id(σStudent.Age<20 (Student⋈Borrow⋈Book)))

i) Find all authors who have written books in multiple genres.

πAuthor.Author_Name(σcount(Genre.Genre_Name)>1 (Wrote⋈Classification⋈Genre⋈Author))

j) Determine all authors who have not written any books classified under the 'Non-Fiction' genre.

πAuthor.Author_Name(Author−(σGenre.Genre_Name='Non-Fiction' (Wrote⋈Classification⋈Genre⋈Author)))

Q2: [10 x 2 = 20 marks]

Consider the following relations:

Student(**snum**: integer, sname: string, major: string, level: string, age: integer)
Class(**name**: string, meets at: string, room: string, **fid***: integer)
Enrolled(**snum***: integer, **cname***: string)
Faculty(**fid**: integer, fname: string, **deptid***: integer)

Bold and underlined represent primary key and fields with * showing foreign keys. The meaning of these relations is straightforward; for example, enrolled has one record per student-class pair such that the student is enrolled in the class.

Write the following queries in SQL. No duplicates should be printed in any of the answers.

A) Find the age of the oldest student who is either a history major or enrolled in a course taught by "B. K. Dar".

```
SELECT      MAX(S.age)
FROM Student S
WHERE       (S.major = 'History')
OR S.snum IN (SELECT E.snum
FROM Class C, Enrolled E, Faculty F
WHERE E.cname = C.name AND C.fid = F.fid
AND F.fname = 'B. K. Dar' )
```

B) Find the names of all classes that either meet in room R128 or have five or more students enrolled.

```
SELECT C.name FROM Class C
WHERE C.room = 'R128'
OR C.name IN (SELECT E.cname
FROM Enrolled E
GROUP BY E.cname
HAVING COUNT (*) >= 5)
```

C) Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

```
SELECT DISTINCT F.fname
        FROM   Faculty F
        WHERE 5 > (SELECT COUNT (E.snum)
                        FROM Class C, Enrolled E
      WHERE C.name = E.cname AND  C.fid = F.fid)
```

D) Find the names of students enrolled in the maximum number of classes.

```
SELECT DISTINCT S.sname
        FROM   Student S
        WHERE          S.snum IN (SELECT E.snum
                              FROM   Enrolled E
                            GROUP BY E.snum
                              HAVING COUNT (*) >= ALL (SELECT COUNT (*)
                                                       FROM   Enrolled E2
                                                     GROUP BY E2.snum ))
```

E) For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR-level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

```
SELECT S.age, S.level
        FROM   Student S
            GROUP BY S.age, S.level,
            HAVING S.level IN (SELECT S1.level
                               FROM   Student S1
                               WHERE S1.age = S.age
                              GROUP BY S1.level, S1.age
                               HAVING COUNT (*) >= ALL (SELECT COUNT (*)
                                                        FROM   Student S2
                                                        WHERE s1.age = S2.age
                                                        GROUP BY S2.level,
                                                        S2.age))
```

The following relations keep track of airline flight information:

Flights(**flno**: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: real)
Aircraft(**aid**: integer, aname: string, cruisingrange: integer)
Certified(**eid***: integer, **aid***: integer)
Employees(**eid**: integer, ename: string, salary: integer)
Note that the Employee relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft, and only pilots are certified to fly. Write each of the following queries in SQL.

F) For each pilot who is certified for more than three aircraft, find the *eid* and the maximum *cruisingrange* of the aircraft for which she or he is certified.

```
SELECT C.eid, MAX (A.cruisingrange)
        FROM   Certified C, Aircraft A
        WHERE C.aid = A.aid
    GROUP BY C.eid
HAVING COUNT (*) > 3
```

G) For all aircraft with *cruisingrange* over 1000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.

```
SELECT Temp.name, Temp.AvgSalary
FROM ( SELECT A.aid, A.aname AS name,
                AVG (E.salary) AS AvgSalary
            FROM   Aircraft A, Certified C, Employees E
            WHERE A.aid = C.aid AND
                C.eid = E.eid AND A.cruisingrange > 1000
GROUP BY A.aid, A.aname ) AS Temp
```

H) Find the *aid*s of all aircraft that can be used on routes from Los Angeles to Chicago.

```
SELECT A.aid
    FROM Aircraft A
    WHERE A.cruisingrange > ( SELECT MIN (F.distance)
                                FROM Flights F
WHERE F.from = 'Los Angeles' AND F.to = 'Chicago' )
```

I) Identify the routes that can be piloted by every pilot who makes more than $100,000.

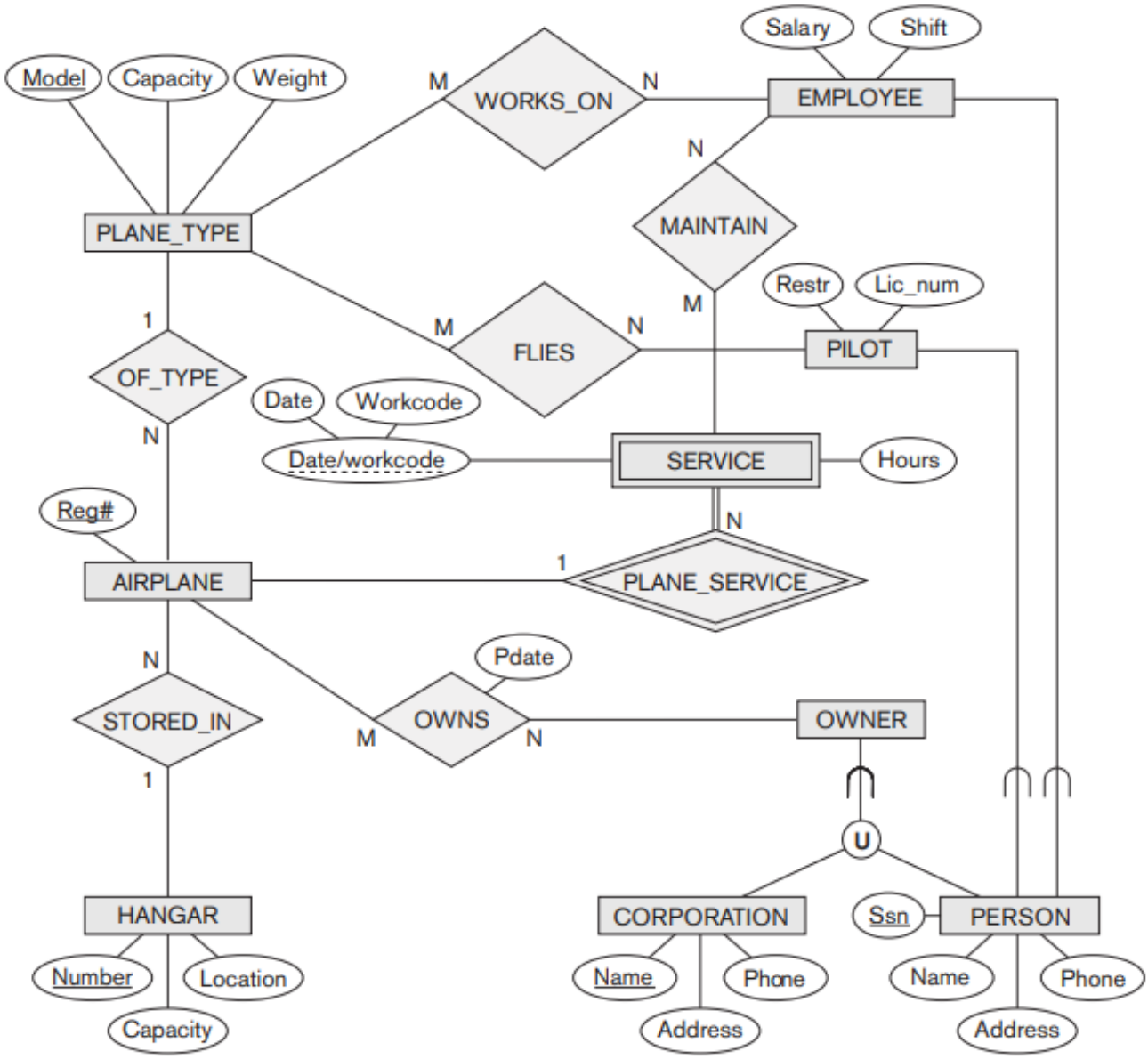
```
SELECT DISTINCT F.from, F.to
        FROM Flights F
        WHERE NOT EXISTS ( SELECT *
                           FROM Employees E
                           WHERE E.salary > 100000
                          AND
                           NOT EXISTS (SELECT *
                                       FROM Aircraft A, Certified C
                                       WHERE A.cruisingrange > F.distance
                                       AND E.eid = C.eid
AND A.aid = C.aid) )
```

J) Print the name and salary of every nonpilot whose salary is more than the average salary for pilots.

```
SELECT E.ename, E.salary
FROM Employees E
WHERE E.eid NOT IN ( SELECT DISTINCT C.eid
FROM Certified C )
AND E.salary > ( SELECT AVG (E1.salary)
FROM Employees E1 WHERE E1.eid IN
( SELECT DISTINCT C1.eid FROM Certified C1 ) )
```

Q3: **[45 marks]**

Consider the following E-ERD and convert it to Relational Schema. Extract tables, primary and foreign keys.



Solution:

1. AIRPLANE (Reg#, OF_TYPE, STORED_IN)

   - Reg# (Primary Key)

   - OF_TYPE (Foreign Key referencing PLANE_TYPE)

   - STORED_IN (Foreign Key referencing HANGAR)

2. PLANE_TYPE (Model, Capacity, Weight)

   - Model (Primary Key)

- Capacity
- Weight

3. HANGAR (Number, Capacity, Location)
   - Number (Primary Key)
   - Capacity
   - Location
4. OWNS (Reg#, Owner_ID, Pdate)
   - Reg# (Foreign Key referencing AIRPLANE)
   - Owner_ID (Foreign Key referencing OWNER)
   - Pdate
5. MAINTAIN (Emp_ID, Service_ID)
   - Emp_ID (Foreign Key referencing EMPLOYEE)
   - Service_ID (Foreign Key referencing SERVICE)
6. PLANE_SERVICE (Reg#, Service_ID)
   - Reg# (Foreign Key referencing AIRPLANE)
   - Service_ID (Foreign Key referencing SERVICE)
7. SERVICE (Service_ID, Date, Hours, Work_code)
   - Service_ID (Primary Key)
   - Date
   - Hours
   - Work_code
8. OWNER (Owner_ID, Type, Ssn, Name, Address, Phone)
   - Owner_ID (Primary Key)
   - Type

- Ssn
- Name
- Address
- Phone

9. PERSON (Ssn, Name, Address, Phone)
   - Ssn (Primary Key)
   - Name
   - Address
   - Phone
10. CORPORATION (Owner_ID, Name, Address, Phone)
    - Owner_ID (Primary Key, Foreign Key referencing OWNER)
    - Name
    - Address
    - Phone
11. PILOT (Ssn, Lic_num, Restr)
    - Ssn (Primary Key, Foreign Key referencing PERSON)
    - Lic_num
    - Restr
12. EMPLOYEE (Emp_ID, Salary, Shift)
    - Emp_ID (Primary Key, Foreign Key referencing PERSON)
    - Salary
    - Shift
13. FLIES (Ssn, Model)
    - Ssn (Foreign Key referencing PILOT)

- Model (Foreign Key referencing PLANE_TYPE)

14. WORKS_ON (Emp_ID, Model)

- Emp_ID (Foreign Key referencing EMPLOYEE)

- Model (Foreign Key referencing PLANE_TYPE)

Part B – ERD – EERD - Attempt all the questions.

Q1: [30 x 0.5 = 15 marks]

Fill the following table using Attribute types from Simple, Composit, derived, multi-values and Data Types from Varchar, DateTime, etc. Please redraw this table on answer sheet fill it.

Sr. No.	Attribute	Attribute Type	SQL Data Type	Sample Value
0	First Name	Simple	Varchar(20)	Zeshan
1	Full Name	Composite	Array or multiple Columns	Zeshan,Khan
2	Roll Number	Composite, Simple	Multiple Varchar(x), Varchar (8)	22, I, 1234, 22I-1234
3	Date of Birth	Composite	Date	20 May 2004
4	Age	Derived	Number, Varchar (8)	20 or 20 Years
5	Semester Grades	Multivalued	Array	A, B, C,D
6	GPA	Derived	Number	3.99
7	Heighest Education	Simple	Varchar(5)	FSc.
8	Is Fee Paid	Simple	Boolean	True, 1
9	Is Degree Completed	Simple	Boolean	True, 1
10	Degree Enrolled	Simple	Varchar(5)	BS(SE)

Some other valid options are also possible and provided marks on thoise.

Q2: [6+1+3+2.5+5+2.5= 20 marks]

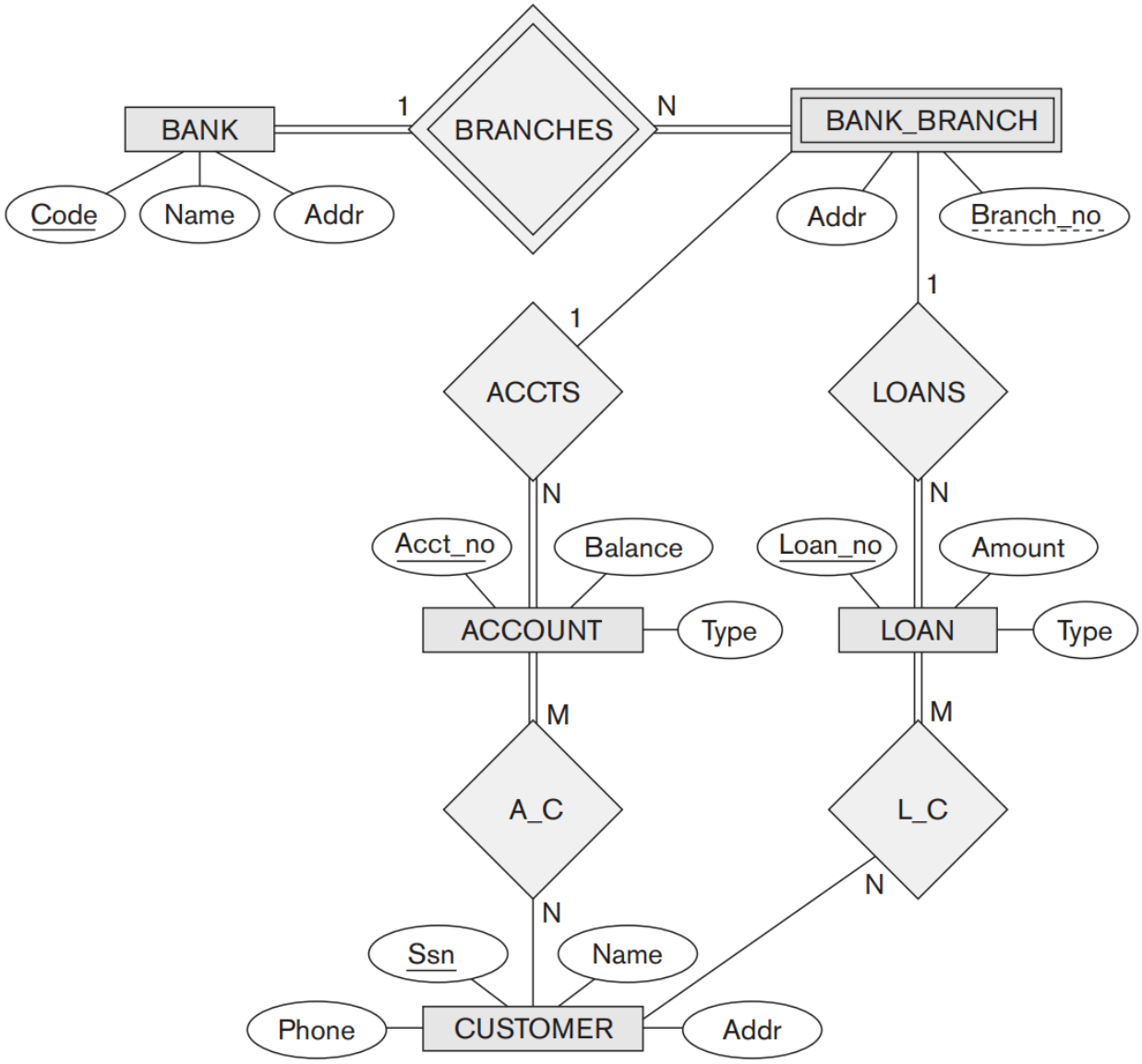
Case Study: Hatfield Recruitment is a job recruitment company which has many client companies who wish them to manage the recruitment process related to job vacancies they

wish to fill. Hatfield Recruitment has several hiring managers, these managers can be allocated any number of such job vacancies, from any of the client companies. Client companies are not allocated to specific hiring managers. Each job vacancy is advertised for a set period and details about the vacancy are released, such as position description, dates between which applications are to be accepted, salary range, location, and a set of prerequisites for the job, such as clean driving licence, UK passport holder, etc. The hiring managers are responsible for identifying a list of suitable applicants from those applying. They will record their details such as name, dob, address and contact number. These applicants will be either existing applicants on record or new applicants. Using the above specification, create a comprehensive ERD diagram that clearly delineates all:

A) Entities [6 entities]
B) Attributes [Some of the attributes are enough]
C) Primary Keys [0.5 for each entity]
D) Relationships/Forighn Keys [0.5 for each relation]
E) Cardinalities [1 for each cardinality (0.5 each side)]
F) Modalities [0.5 for each modality]

Q3: [15 marks]

Consider the BANK ER schema in Figure below, and suppose that it is necessary to keep track of different types of ACCOUNTS (SAVINGS_ACCTS, CHECKING_ACCTS, …) and LOANS (CAR_LOANS, HOME_LOANS, …). Suppose that it is also desirable to keep track of each ACCOUNT's TRANSACTIONS (deposits, withdrawals, checks, …) and each LOAN's PAYMENTS; both of these include the amount, date, and time. Modify the BANK schema, using ER and **EER** concepts of specialization and generalization. State any assumptions you make about the additional requirements.

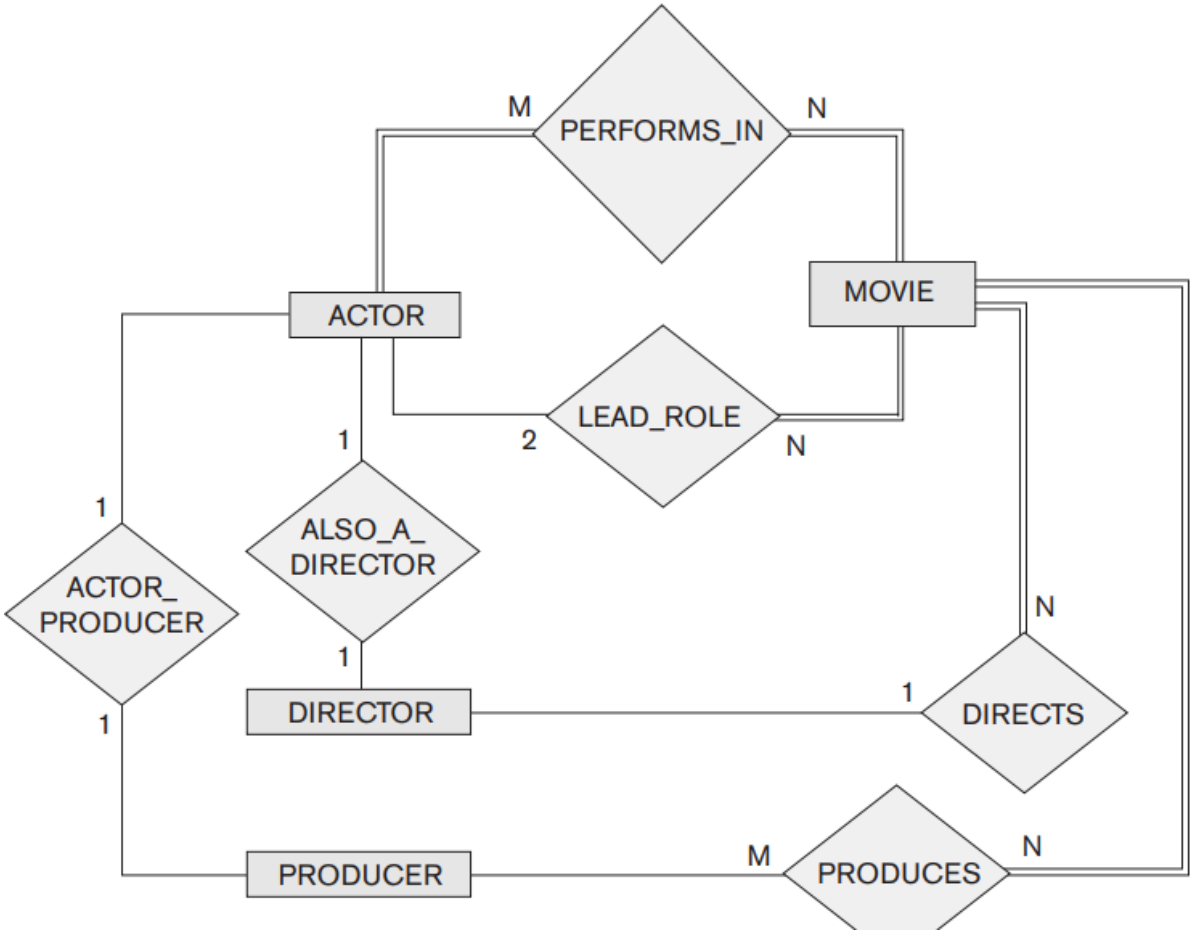


Some of the Enhanced ERDs e.g. Account and Loan can be generalized and then types can be specialized types.

Q4:

[0.7*14=10 marks]

Consider the ER schema for the MOVIES database in Figure below.



Assume that MOVIES is a populated database. ACTOR is used as a generic term and includes actresses. Given the constraints shown in the ER schema, respond to the following statements with ***True, False***, or ***Maybe***. Assign a response of ***Maybe*** to statements that, although not explicitly shown to be ***True***, cannot be proven ***False*** based on the schema as shown. Justify each answer.
[0.7 for each correct answer, rounded up to 0.5]

i. i. There are no actors in this database that have been in no movies. True

ii. ii. There are some actors who have acted in more than ten movies. Maybe

iii. iii. Some actors have done a lead role in multiple movies. Maybe

iv. iv. A movie can have only a maximum of two lead actors True

v. v. Every director has been an actor in some movie. Maybe

vi. vi. No producer has ever been an actor. Maybe

vii. vii. A producer cannot be an actor in some other movie. False

viii. viii. There are movies with more than a dozen actors. Maybe

ix. ix. Some producers have been a director as well. Maybe

x. x. Most movies have one director and one producer. Maybe

xi. xi. Some movies have one director but several producers. Maybe

xii. xii. There are some actors who have done a lead role, directed a movie, and produced a movie. Maybe

xiii. xiv. No movie has a director who also acted in that movie. Maybe

Part C – Normalization and Transaction Management

Question 1. [30 Marks]

(a). Please write the correct option on the answer sheet. Clearly write the part number along with the correction option. In case of cutting, overwriting and no attempt on the answer sheet, the question will be marked zero.

Cutting and overwriting was marked zero. No attempt on the answer book is marked zero. Missing is marked zero. For MCQ 2, regardless of the answer 1 mark is given.

1. **D**
2. **D (mark is given for any option)**
3. **A**
4. **D**
5. **B**
6. **D**
7. **A**
8. **B**
9. **A**
10. **B**
11. **A**
12. **B**
13. **D**
14. **C**
15. **D**
16. **D**
17. **B**
18. **C**
19. **A**
20. **B**
21. **A**
22. **D**
23. **C**
24. **B**
25. **D**
26. **C**
27. **C**
28. **D**
29. **A**
30. **C**

Question 2 **[Marks 10+5+15 = 30]**

Below are two database transactions. Assume that they are concurrent transactions. Please answer the following questions.

Time	T1	T2	A	B
1	Start transaction T1		**100**	**100**
2	Read the value of item A			
3	Perform operations A = A * 20			
4	Write A			
5	Read the value of item B	Start transaction T2		
6	Perform operations B = A + B	Read the value of item B		
7	Write B	Perform operations B = B - 10		
8	..	Write B		
9	..	Read the value of item A		
10	..	Perform operations A = A - 20 + B		
11		Write A		
12	..	Commit transaction T2		
13	Roll Back transaction T1			

1. **What will be value of A and B after the execution of the above transaction T1 and T2. Draw the table on answer sheet to show complete values of A and B along all-time stamps. Exact following transaction table is required. If mistakes, it is marked zero.**

Time	T1	T2	A	B
1	Start transaction T1		100	100
2	Read the value of item A		100	100
3	Perform operations A = A * 20		100	100
4	Write A		2000	100
5	Read the value of item B	Start transaction T2	2000	100
6	Perform operations B = A + B	Read the value of item B	2000	100
7	Write B	Perform operations B = B - 10	2000	2100
8	..	Write B	2000	90
9	..	Read the value of item A	2000	90
10	..	Perform operations A = A - 20 + B	2000	90
11		Write A	2070	90
12	..	Commit transaction T2	2070	90
13	Roll Back transaction T1		100	100

2. ***What type of anomaly does this transaction presents? Please explain the reasoning***

Uncommitted dependency problem **.** ***if answer with reasoning is provided then 5 marks. If only anomaly name then 2 marks else zero,***

3. ***Perform two-phase locking and execute the above transaction.*** **Draw the table on answer sheet to show complete values of A and B along all-time stamps. Also write the updated values for A and B after the transaction is executed.**

Exact table is require with all time stamps. All locks along with the values. It is marked either full or zero.

Time	T1	T2	A	B
1	Start transaction T1		100	100
2	Write lock A		100	100
3	Read the value of item A		100	100
4	Perform operations A = A * 20		100	100
5	Write A		2000	100
6	Write Lock B		2000	100
7	Read the value of item B	Start transaction T2	2000	100
8	Perform operations B = A + B	Write Lock B	2000	100
9	Write B	WAIT	2000	2100
10	..	WAIT	2000	2100
11	..	WAIT	2000	2100
12	..	WAIT	2000	2100
13		WAIT	2000	2100
14	..	WAIT	2000	2100
15	Roll Back transaction T1 / Unlock A and B	WAIT	100	100
16		Read the value of item B	100	100
17		Perform operations B = B – 10	100	100
18		Write B	100	90
19		Write Lock A	100	90
20		Read the value of item A	100	90
21		Perform operations A = A – 20 + B	100	90
22		Write A	170	90
23		Commit transaction T2 Unlock A and B	170	90

Question 3 [20 Marks]

Consider the table given below containing the details of the doctor, patient, and their appointments. The data is being identified with unique appointment number (Appt no). The system stores the patient's name and phone number. Patients can take appointments with the doctors and based on their appointment types; their duration is determined. The system also stores doctors first names with their IDs. Based on the information provided. Please answer the following questions

All parts are either marked full or zero

a. Is the following relation in 1 NF? Answer in yes or no. [2]

Yes

b. If the answer of **part a** is NO, then convert the relation into 1 NF and write the schema of 1 NF relations. If the relation is already in 1 NF, then write "Already in 1NF". [2]

Already in 1NF

c. Identify the functional dependencies from the relation mentioned in **part b**. [3]

All functional dependencies on the apptno

d. Is/Are the relation(s) mentioned in answer of **part b** in 2NF? Answer in YES or NO. [2]

Yes

e. Convert the relation into 2NF if the answer of **part c** is NO. If the relation is already in 2 NF, then write "Already in 2NF". [2]

Already in 2NF

f. Are the relations mentioned in **part f** in 3 NF? (YES or NO) [2]

No

g. If the answer of **part g** is NO, then convert the relations into 3 NF. If the relation is already in 3 NF, then write "Already in 3NF". [4]

Appointment Table

Appt No	Appt Date	Appt Time	Appt Type	Patient ID	Doctor ID
1	12/1/2015	3:00 AM	Physical	466927	C678
2	12/1/2015	3:00 AM	Shot	456789	A528
3	12/1/2015	3:15 AM	Flu	194756	S626
4	12/2/2015	10:00 AM	Migraine	329657	A528
5	12/2/2015	10:15 AM	Shot	987453	G123
6	12/2/2015	10:30 AM	Shot	384788	S626
7	12/2/2015	10:45 AM	Flu	438754	C678
8	12/2/2015	11:00 AM	Physical	345875	A528
9	12/3/2015	10:30 AM	Physical	466927	C678
10	12/3/2015	9:00 AM	Migraine	345875	C678

Patient Table

Patient ID	First Nm	Last Nm	Phone
194756	Brandon	Pierre	432-7877
329657	Marcus	Schwartz	239-5502
345875	Carla	Basich	857-5566
384788	Tonya	Johnson	432-8806
438754	Iliana	Hnatt	823-4303
456789	Sue	Carey	432-1234
466927	Lisa	Garcia	562-3456
987453	Mike	Jones	456-0202

Doctor Table

Doctor ID	Doctor Nm
A528	Lopez
C678	Chapman
G123	Gray
S626	Smith

Appt Type Table

Appt Type	Planned Duration
Flu	0.50
Migraine	0.50
Physical	1.00
Shot	0.25

h. Is the resultant 3NF schema in BCNF? [1]

NO

i. Write the final normalized schema in the space below. Make sure to define all primary keys and foreign keys [2]

Appointment Table

Appt No	Appt Date	Appt Time	Appt Type	Patient ID	Doctor ID
1	12/1/2015	3:00 AM	Physical	466927	C678
2	12/1/2015	3:00 AM	Shot	456789	A528
3	12/1/2015	3:15 AM	Flu	194756	S626
4	12/2/2015	10:00 AM	Migraine	329657	A528
5	12/2/2015	10:15 AM	Shot	987453	G123
6	12/2/2015	10:30 AM	Shot	384788	S626
7	12/2/2015	10:45 AM	Flu	438754	C678
8	12/2/2015	11:00 AM	Physical	345875	A528
9	12/3/2015	10:30 AM	Physical	466927	C678
10	12/3/2015	9:00 AM	Migraine	345875	C678

Patient Table

Patient ID	First Nm	Last Nm	Phone
194756	Brandon	Pierre	432-7877
329657	Marcus	Schwartz	239-5502
345875	Carla	Basich	857-5566
384788	Tonya	Johnson	432-8806
438754	Iliana	Hnatt	823-4303
456789	Sue	Carey	432-1234
466927	Lisa	Garcia	562-3456
987453	Mike	Jones	456-0202

Doctor Table

Doctor ID	Doctor Nm
A528	Lopez
C678	Chapman
G123	Gray
S626	Smith

Appt Type Table

Appt Type	Planned Duration
Flu	0.50
Migraine	0.50
Physical	1.00
Shot	0.25