

## Question 1 [10 marks]

Define  $R \Delta S$  (**R Symmetric Difference S**) as the relation that contains **tuples that belong to either R or S, but not both**. Assume that both R and S have the same schema, (A, B, C), and are "Symmetric Difference Compatible," wherever applicable:

**Question 1.1 Can you express  $R \Delta S$  using basic relational operations?**

**If yes, provide the expression. If not, explain why and what is missing.**

$$R \Delta S = (R - S) \cup (S - R)$$

OR

$$R \Delta S = (R \cup S) - (R \cap S)$$

**Question 1.2 Can you express  $R \Delta S$  in SQL?**

**If yes, provide the SQL statement. If not, explain why and what is missing.**

```
(SELECT * FROM R
WHERE (R.A, R.B, R.C) NOT IN (SELECT * FROM S))
UNION
(SELECT * FROM S
WHERE (S.A, S.B, S.C) NOT IN (SELECT * FROM R));

(SELECT * FROM R EXCEPT SELECT * FROM S)
UNION
(SELECT * FROM S EXCEPT SELECT * FROM R);

(SELECT * FROM R
UNION
SELECT * FROM S)
EXCEPT
(SELECT * FROM R
INTERSECT
SELECT * FROM S);
```

## Schema (Used for Both RA & SQL Questions)

We consider a **Hospital Management System** with the following **five** relations:

- **Patient** (PID:number, PName:string, Age:number, Gender:string, Address:string)  
*Each patient has a unique ID and personal details.*
- **Doctor** (DID:number, DName:string, Specialization:string, Experience:number, Department:string)  
*Each doctor has a unique ID, a specialization, years of experience, and a department.*
- **Appointment** (AID:number, PID:number, DID:number, Date:date, Time:string, Status:string)  
*Each appointment has a unique ID, links a patient to a doctor, includes a date and time, and has a status (Scheduled, Completed, Cancelled).*
- **Treatment** (TID:number, PID:number, DID:number, TreatmentType:string, StartDate:date, EndDate:date, Cost:number)  
*Each treatment has a unique ID, is linked to a patient and doctor, includes a type (Surgery, Therapy, Medication), duration, and cost.*
- **Billing** (BID:number, PID:number, TotalAmount:number, PaymentStatus:string, BillingDate:date)  
*Each billing record has a unique ID, is associated with a patient, contains the total amount, payment status (Paid, Pending, Overdue), and billing date.*

## Question 2 Relational Algebra [20 marks]

Answer the following queries in Relational Algebra.

If the query cannot be expressed in Relational Algebra, briefly explain the reason for it.

Only use the basic RA operators taught in lectures ( $\sigma$ ,  $\Pi$ ,  $\bowtie$ ,  $\bowtie_\theta$ ,  $\cup$ ,  $-$ ,  $\cap$ ,  $\times$ ,  $\rho$ ,  $/$ ,  $\div$ )

**Question 2.1 [3 marks]** List all doctors (names and IDs) who have **never scheduled an appointment**.

DoctorsWithAppointments =  $\Pi$  DID, DName (Doctor  $\bowtie$  Appointment)

Result =  $\Pi$  DID, DName (Doctor) - DoctorsWithAppointments

Result

**Question 2.2 [5 marks]** Retrieve the names and IDs of patients who **have received at least one treatment from an experienced doctor** (Experience > 10) and **have also undergone a high-cost treatment** (Cost > 5000).

```
ExperiencedDoctors = σ Experience > 10 (Doctor)
HighCostTreatments = σ Cost > 5000 (Treatment)
PatientsWithExperiencedDoctors = π PID (Patient ⋈ Treatment ⋈ ExperiencedDoctors)
PatientsWithHighCostTreatment = π PID (Patient ⋈ HighCostTreatments)
Result = π PID, PName (Patient ⋈ (PatientsWithExperiencedDoctors ∩
PatientsWithHighCostTreatment))
Result
```

**Question 2.3 [5 marks]** Retrieve the names and IDs of patients who **have taken treatment from all doctors** in the **Cardiology department**.

```
CardiologyDoctors = π DID(σ Department='Cardiology'(Doctor))
PatientDoctorPairs = π PID, DID (Treatment)
Result = π PID, PName (Patient ⋈ (PatientDoctorPairs ÷ CardiologyDoctors))
Result
```

**Question 2.4 [5 marks]** Retrieve the *names and IDs of doctors* who have treated **at least two different patients** with the **same treatment type** and **at least one of those patients has paid bills**.

$T1 = \rho T1 \text{ (Treatment)}$

$T2 = \rho T2 \text{ (Treatment)}$

$\text{SameTreatmentTwoPatients} = \sigma T1.\text{TreatmentType} = T2.\text{TreatmentType} \wedge T1.\text{PID} \neq T2.\text{PID} (T1 \times T2)$

$\text{DoctorForTreatment} = \pi T1.\text{DID}, T1.\text{PID} (\sigma T1.\text{DID} = T2.\text{DID} \text{ SameTreatmentTwoPatients})$

$\text{PatientsWithPaidBills} = \pi \text{PID} (\sigma \text{PaymentStatus} = \text{'Paid'} \text{ (Billing)})$

$\text{ValidDoctorForTreatment} = \text{DoctorForTreatment} \bowtie \text{PatientsWithPaidBills}$

$\text{Result} = \pi T1.\text{DID}, \text{Doctor.DName} (\text{ValidDoctorForTreatment} \bowtie \text{Doctor})$

$\text{Result}$

**Question 2.5 [2 marks]** Find the doctors (names and IDs) who has treated the **Maximum Number of Patients**.

Cannot be done in RA since RA does not allow aggregate functions.

## Question 3 SQL [20 marks]

Write the following queries in SQL using **SQLite3 syntax**. Ensure that your queries return the correct results based on the **Hospital Management System database**.

**Question 3.1 [5 marks]** Find doctors (list names, IDs and experience) who have **never treated a patient** and **order by experience**.

```
SELECT D.DID, D.DName, D.Experience
FROM Doctor D
LEFT JOIN Treatment T ON D.DID = T.DID
GROUP BY D.DID, D.DName, D.Experience
HAVING COUNT(T.PID) = 0
ORDER BY D.Experience DESC;
--or
SELECT D.DID, D.DName, D.Experience
FROM Doctor D
LEFT JOIN Treatment T ON D.DID = T.DID
WHERE T.DID IS NULL
ORDER BY D.Experience DESC;
--or
SELECT D.DID, D.DName, D.Experience
FROM Doctor D
WHERE D.DID NOT IN (SELECT D.DID FROM DOCTOR D, Treatment T ON D.DID = T.DID)
ORDER BY D.Experience DESC;
```

**Question 3.2 [5 marks]** Find **all patients** (names, IDs) and **their treatments** (treatment type and IDs), **including those without a treatment** and **treatments without a patient record**.

```
SELECT P.PID, P.PName, T.TID, T.TreatmentType, T.DID
FROM Patient P
LEFT JOIN Treatment T ON P.PID = T.PID

UNION ALL

SELECT T.PID, P.PName, T.TID, T.TreatmentType, T.DID
FROM Treatment T
LEFT JOIN Patient P ON P.PID = T.PID
WHERE P.PID IS NULL;
```

**Question 3.3 [5 marks]** Retrieve the names and IDs of patients who **have taken treatment from all doctors** in the **Cardiology department**.

```
SELECT P.PID, P.PName
FROM Patient P
WHERE NOT EXISTS (
    SELECT D.DID
    FROM Doctor D
    WHERE D.Department = 'Cardiology'
    EXCEPT
    SELECT T.DID
    FROM Treatment T
    WHERE T.PID = P.PID
);
--Or
SELECT P.PID, P.PName
FROM Patient P
JOIN Treatment T ON P.PID = T.PID
JOIN Doctor D ON T.DID = D.DID
WHERE D.Department = 'Cardiology'
GROUP BY P.PID, P.PName
HAVING COUNT(DISTINCT D.DID) = (SELECT COUNT(*) FROM Doctor WHERE Department = 'Cardiology');
```

**Question 3.4 [5 marks]** Find the Doctor Who Has Treated the **Maximum Number of Patients**.

```
SELECT Doctor.DID, Doctor.DName, COUNT(DISTINCT Treatment.PID) AS PatientCount
FROM Doctor
JOIN Treatment ON Doctor.DID = Treatment.DID
GROUP BY Doctor.DID, Doctor.DName
HAVING COUNT(DISTINCT Treatment.PID) = (
    SELECT MAX(PatientCount)
    FROM (
        SELECT COUNT(DISTINCT PID) AS PatientCount
        FROM Treatment
        GROUP BY DID
    ) AS Subquery
);
--or
SELECT D.DID, D.DName
FROM Doctor D
JOIN Treatment T ON D.DID = T.DID
GROUP BY D.DID, D.DName
HAVING COUNT(T.PID) = (
    SELECT MAX(PatientCount)
    FROM (SELECT COUNT(PID) AS PatientCount FROM Treatment GROUP BY DID) AS Temp
--or
WITH DoctorPatientCount AS (
    SELECT Treatment.DID, Doctor.DName, COUNT(DISTINCT Treatment.PID) AS PatientCount
    FROM Treatment
    JOIN Doctor ON Treatment.DID = Doctor.DID
    GROUP BY Treatment.DID, Doctor.DName
)
SELECT DID, DName, PatientCount
FROM DoctorPatientCount
WHERE PatientCount = (SELECT MAX(PatientCount) FROM DoctorPatientCount);
```