1. A Research Group at IIIT-Hyderabad consists of Researchers who lead Projects, develop Algorithms, and utilize Datasets. The figure provided on the **next page** represents the preliminary conceptual schema (ER Diagram).

Problem Statement: The Lab Director has released the following **3 operational requirements**. Modify the model (by drawing directly on the Figure on the next page) to support these requirements.

Instructions:

- Interpret the requirements and select the most appropriate modeling constructs.
- **Make and state reasonable assumptions only if a constraint is not explicitly defined** and necessary.
- Please follow the standard notations of ER diagram.

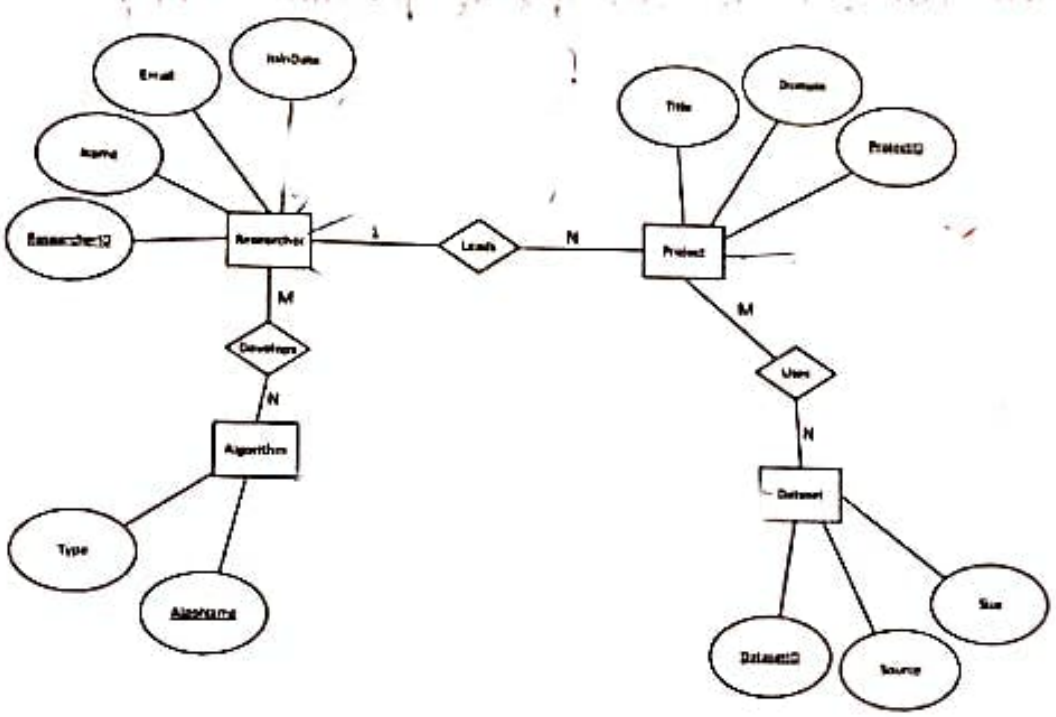
(a) [2 points] **Mentorship Hierarchy**
We need to model the internal reporting structure. Every Junior Researcher is assigned a mentor, who is also a Researcher in the lab. A Junior Researcher is required to have **exactly one** mentor, but a Senior Researcher can supervise **multiple** juniors (or none at all).

(b) [2 points] **Experimental Logs**
The current model tracks who "Develops" algorithms, but it does not track actual experiments. We need to log every instance where a **Researcher runs a specific Algorithm** on a specific **Dataset**. For every such run, we must record the resulting `Accuracy_Score`.

(c) [2 points] **Funding Installments**
Projects do not receive money in a lump sum; they receive it in installments. An `Installment` cannot exist without being linked to a Project. Furthermore, installments are identified by a simple `Serial_No` (e.g., Installment 1, 2) which repeats across different projects, along with the `Amount` received.

ER Diagram Workspace

(Please modify this diagram as per the requirements)



2 [6 points] Answer the following. Each sub-part is of 2 points.

(a) What is meant by procedural and non-procedural DML queries? Is SQL a procedural or a non-procedural language?

(b) Consider the following database schema:

- **Student(sid, roll, email, dept_id)**
  - sid = auto-increment integer
  - roll = unique within the university
  - email = unique but can be NULL
  - dept_id = department of the student
- **Department(dept_id, dept_name, hod_id)**
  - dept_id = unique
  - dept_name may repeat across departments
  - hod_id = faculty ID of head of department
- **Course(cid, cname, dept_id)**
  - cid = unique
  - each course belongs to exactly one department
- **Enroll(sid, cid, semester, grade)**
  - A student can take many courses
  - A course can have many students
  - A student can enroll in the same course again in another semester

Question: Which of the following statements about keys are TRUE for the above schema? Circle the correct options. No additional explanation is required. (Multiple options maybe correct)

1. `sid` in `Student` is a surrogate key.
2. In `Student`, {roll, email} is a super key.
3. `dept_id` in `Course` is a foreign key referencing `Department(dept_id)`.
4. In `Department`, {dept_id, dept_name} is a candidate key.

(c) Consider the following database schema and the constraint described below.

- `Department(dept_id INT, dname VARCHAR, budget INT)`
- `Employee(eid INT, name VARCHAR, dept_id INT, salary INT)`

Constraint C: Every department must have at least one employee. Formally:

You are given the following SQL statements. Which of them **definitely violate** Constraint C? (Multiple options maybe correct)

1.
```
INSERT INTO Department(dept_id, dname, budget)
VALUES (50, 'Operations', 800000);
```

2.
```
DELETE FROM Employee
WHERE dept_id = 20;
```

3.
```
UPDATE Employee
SET dept_id = 99
WHERE eid = 5;
```

4.
```
DELETE FROM Department
WHERE dept_id = 10;
```

3. [6 points] Prove that, if a relation R is in 3NF and every key is simple , then R is in BCNF.

4 [2 points] Consider a table Deliveries that tracks shipment data for a logistics company.

```
driverID (Integer)
packageID (Integer, Unique)
distance (Integer)
```

You need to find the driver-id of the driver who has covered the highest TOTAL distance across all their deliveries. Assume there is a single driver with the unique maximum total. Which of the following queries will correctly obtain this result? (Multiple options maybe correct)

a) SELECT driverID FROM Deliveries WHERE SUM(distance) = (SELECT MAX(SUM(distance)) FROM Deliveries GROUP BY driverID);

b) SELECT driverID FROM Deliveries GROUP BY driverID ORDER BY SUM(distance) DESC LIMIT 1;

c) SELECT driverID FROM Deliveries GROUP BY driverID HAVING SUM(distance) >= ALL (SELECT SUM(distance) FROM Deliveries GROUP BY driverID);

d) SELECT driverID FROM Deliveries GROUP BY driverID HAVING SUM(distance) = MAX(distance);

5. [4 points] Consider the following three relations R, S, and T.

Table R

A	B
a1	10
a2	10
a2	30
a3	40

Table S

B	C
100	5
200	25
300	35
400	50

Table T

A	C
a1	5
a2	5
a2	50
a3	25

Evaluate the result of the following Relational Algebra expression. Show the final schema and the set of tuples in the result.

$$\pi_{A,C}\left(\sigma_{C<30}(R \bowtie_{R.B \geq S.C} S)\right) - \pi_{A,C}\left(\sigma_{C=5}(R \bowtie_{R.A=T.A} T)\right)$$

6 [4 points] In SQL, standard aggregation (using GROUP BY) allows you to calculate totals for specific groups. However, when performing multi-dimensional analysis, we often need to see subtotals for all possible combinations of the grouping columns.

The CUBE operator generates a result set that shows aggregates for all permutations of values in the selected columns.

Consider a table named Sales. Below is the transformation from raw data to a "Cubed" result set. Note how the output contains rows for specific combinations (Red/L) as well as summary rows (All Red, All Large, and Total).

Input Table (Sales)

Color	Size	Num
Red	L	10
Red	S	20
Blue	L	30

Target Output (CUBE Operation)

Color	Size	Sum(Num)
Red	L	10
Red	S	20
Blue	L	30
Red	ALL	30
Blue	ALL	30
ALL	L	40
ALL	S	20
ALL	ALL	60

Write a SQL query that produces the Target Output shown above using the Sales table without using the CUBE operator. Write a query specific to the given table Sales.

7. Consider the relation $R(A, B, C, D, E, F)$ with the following set of Functional Dependencies:

1. $A \rightarrow B$
2. $B \rightarrow A$
3. $AC \rightarrow D$
4. $D \rightarrow CE$

Answer the following:

(a) [3 points] Identify all **Candidate Keys** for this relation. Based on these keys, state the **Normal Form** of the relation and justify your answer.

(b) [3 points] Given the functional dependencies above, evaluate the validity of the following *new* dependencies.

- If valid, state "Valid".
- If invalid, make it valid by **truncating attributes from the Right-Hand Side (RHS) minimally**. (You may strictly only remove attributes from the RHS).

i. $BD \rightarrow AEF$

ii. $AE \rightarrow BC$

iii. $ABF \rightarrow DF$

8. [5 points] Suppose we have three relations r (A, B), s(B, C), and t(B, D), with all attributes declared as not null. Consider the expressions

- r natural left outer join (s natural left outer join t)
- (r natural left outer join s) natural left outer join t

a) Give instances of relations r, s and t such that in the result of the second expression, attribute C has a null value but attribute D has a non-null value.

b) Is the above pattern, with C null and D not null possible in the result of the first expression? Explain why or why not.

9. Consider the following relational schema of a hospital database:

Patient(pid, name, age, city)
Visit(vid, pid, doctor, diagnosis, fees)
Medicine(mid, diagnosis, med_name, duration)

The hospital wants to track **"expensive-case patients"**, patients whose **highest-fee visit among all visits costing more than Rs. 1500** should be recorded, along with the recommended medicine for the diagnosis in that visit. If a patient has multiple such high-fee visits, only the visit with the **maximum fee (among fees > 1500)** should be considered.

(a) [3 points] Create a view named `ExpensiveCases` containing the following attributes:

- patient name
- patient age
- diagnosis of the highest-fee visit (among visits where fees 1500)
- the highest visit fee
- the recommended medicine name

The view must contain at most one row per patient.

(b) [1 point] The hospital updates its policy for patients living in Hyderabad. If the fee in the view is ≤ 2000, then the fee should be changed to `NULL`. Write the SQL statement to update the `ExpensiveCases` view to reflect this change. **Do not modify the base tables directly; perform the update on the view.**

(c) [1 point] A diagnosis named `'Migraine'` was found to be incorrect, and the hospital wants to remove all such rows from the `ExpensiveCases` view. Write the SQL statement to delete these rows from the view.