Experiment No.1
Design an EntityRelationship (ER) / Extended Entity-Relationship (EER) Model.
Date of Performance:
Date of Submission:

Aim :- Identify the case study and detailed statement of the problem. Design an EntityRelationship (ER) / Extended Entity-Relationship (EER) Model.

Objective :- To identify and explore a real world problem, and to design an Entity Relationship (ER) / Extended Entity-Relationship (EER) Model.

Theory:

1. **Entity:**

- An entity is a real-world object or concept that exists independently and has distinguishable attributes.
- In a database context, an entity represents a table, and each row in that table represents a unique instance of that entity.
- For example, in a university database, entities could include Student, Course, Professor, Department, etc.
- Each entity has a set of attributes that describe its properties.

2. **Attributes:**

- Attributes are the properties or characteristics that describe an entity.
- They represent the data we want to store about each instance of an entity.
- For example, attributes of a Student entity might include StudentID, Name, Age, GPA, etc.
- Attributes can be categorized as simple (atomic) attributes, which cannot be divided further, or composite attributes, which are made up of smaller sub-parts.

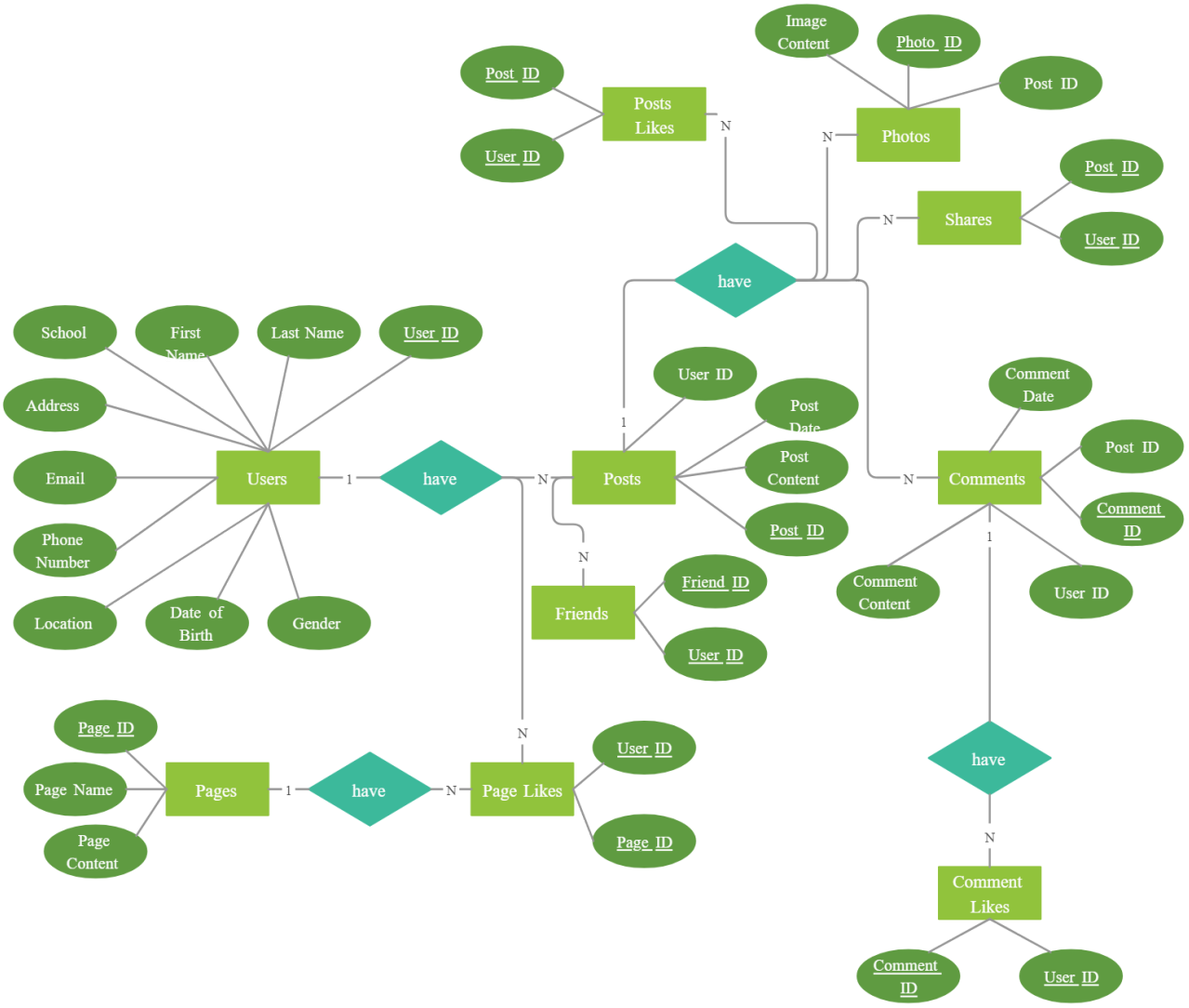
3. **Relationships:**

- Relationships describe how entities are related to each other or how they interact.
- They represent the associations between entities.
- Relationships are depicted as lines connecting related entities in the ER diagram.
- Each relationship has a degree, indicating the number of entities involved. It could be unary (involving one entity), binary (involving two entities), or ternary (involving three entities).
- Relationships also have cardinality, which defines the number of instances of one entity that can be associated with the number of instances of another entity through the relationship.

4. Cardinality:

- Cardinality specifies the number of instances of one entity that are related to the number of instances of another entity through a relationship.
- It defines the maximum and minimum number of occurrences of one entity that can be associated with the occurrences of another entity.
- Common cardinality constraints include:
  I. One-to-One (1:1): Each instance of one entity is associated with exactly one instance of another entity, and vice versa.
  II. One-to-Many (1:N): Each instance of one entity is associated with zero or more instances of another entity, but each instance of the second entity is associated with exactly one instance of the first entity.
  III. Many-to-One (N:1): The reverse of One-to-Many; many instances of one entity are associated with one instance of another entity.
  IV. Many-to-Many (N:N): Many instances of one entity can be associated with many instances of another entity.

Implementation:



Conclusion:

1. **Define Entity, Attributes(also types) and Relationship between entities**

a) Entity: An entity is a real-world object or concept that exists independently and is distinguishable from other objects. In the context of databases and information systems, an entity typically corresponds to a table in a database. For example, in a database for a library system, entities could include "Book," "Author," "Member," etc.

b) Attributes: Attributes are the properties or characteristics that describe an entity. They represent the specific pieces of information associated with an entity. Attributes have types, which define what kind of data can be stored in them. For instance, in the "Book" entity, attributes could include "Title" (a string), "ISBN" (a unique identifier), "Publication Year" (an integer), and so on.

c) Relationships: Relationships define how entities are connected or associated with each other. They describe the interactions and dependencies between entities in a database. Relationships can be one-to-one, one-to-many, or many-to-many. In a library database, for example, there would likely be a relationship between the "Book" entity and the "Author" entity, where one author can have written many books (one-to-many), or vice versa, depending on the design of the system.

2. Write ER/EER diagram notations

a) Entity: Represented by a rectangle with the entity name written inside.
b) Attribute: Represented by an oval or ellipse connected to its respective entity by a line. The attribute name is written inside the oval.
c) Primary Key: Indicated by underlining the attribute(s) that uniquely identify each entity instance.
d) Composite Attribute: An attribute that can be further divided into smaller sub-parts. Represented by an oval with multiple smaller ovals inside, each representing a sub-part, connected to the main oval by lines.
e) Derived Attribute: An attribute whose value can be calculated from other attributes. Represented by an oval with a dashed line connecting it to the attributes it is derived from.
f) Multivalued Attribute: An attribute that can hold multiple values for a single entity instance. Represented by a double oval.
g) Relationship: Represented by a diamond shape connecting two entities, with lines extending from it to each participating entity. The relationship name is written inside the diamond.
h) Cardinality: Represented near the ends of the relationship lines to indicate how many instances of one entity can be associated with instances of the other entity. Common notations include "1" for one, "M" for many, and "0" for optional.
i) Weak Entity: Represented with a double rectangle, with the identifying relationship connecting it to its owner entity.
j) ISA (Inheritance): Represented by a triangle, with the parent entity at the top and the child entities at the bottom, connected by lines.
k) Key Attribute: Represented by underlining the attribute that serves as the primary identifier within a subclass entity.