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

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.

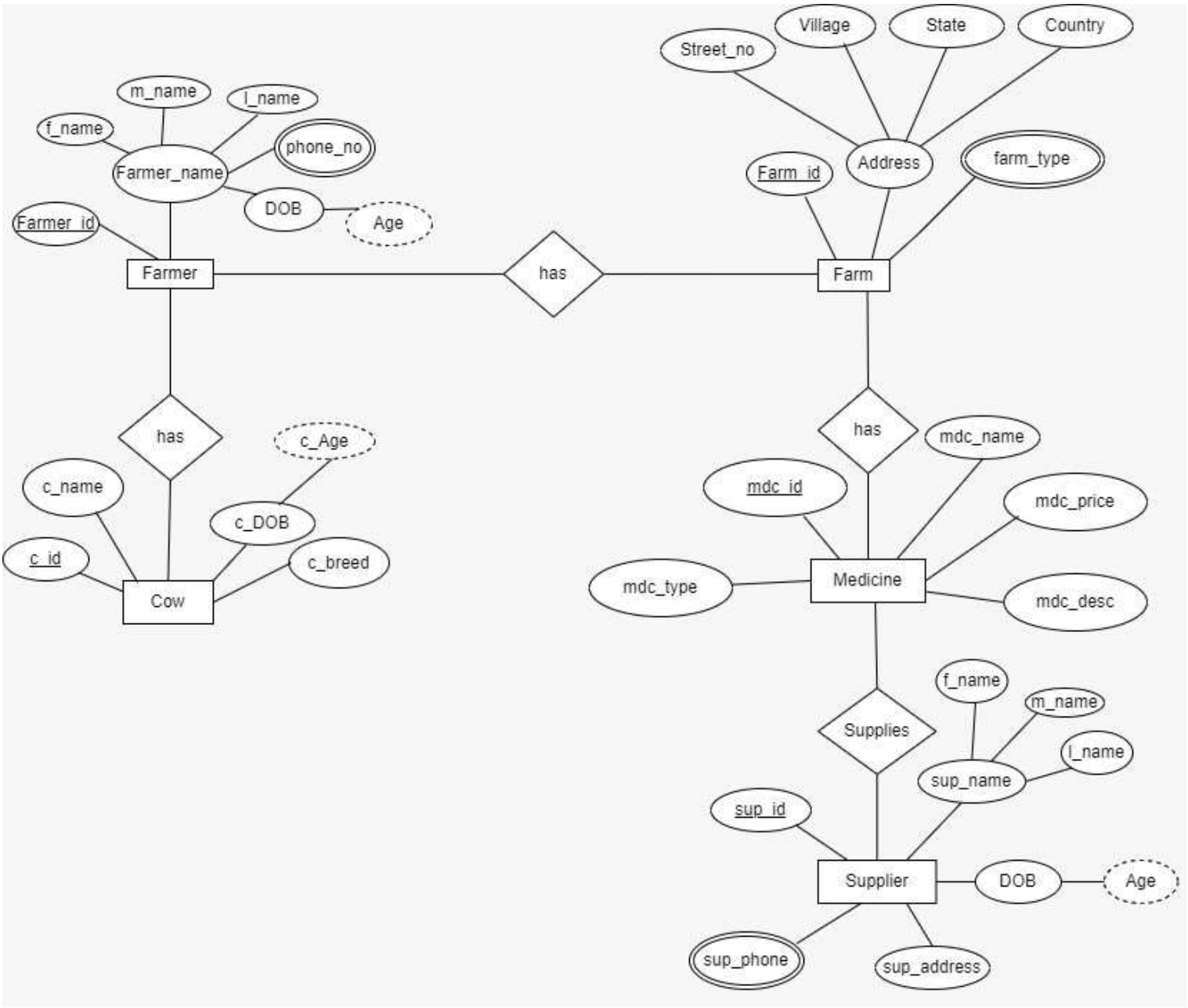
II. One-to-Many (1:N): Each instance of one entity is associated with zero or more instances of another entity, but each instance of the second entity is associated with exactly one instance of the first entity.

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:

Farmers Management System


Village
State
Country
Street_no
m_name
l_name
f_name
phone_no
Farmer_name
DOB
Age
Farmer_id
Farm_id
Address
farm_type
Farmer
has
Farm
has
c_Age
has
mdc_name
c_name
mdc_id
mdc_price
c_DOB
c_id
c_breed
mdc_type
Medicine
mdc_desc
Cow
f_name
m_name
Supplies
l_name
sup_name
sup_id
Supplier
DOB
Age
sup_phone
sup_address

Conclusion:

In conclusion, the Entity-Relationship (ER) diagram for the Farmers Management System provides a comprehensive overview of the system's structure and relationships between various entities. Through this diagram, we have identified the key entities such as farmers, farm, cow, medicine and suppliers, along with their attributes and relationships.

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

Ans. **Entity:** An entity is a distinct object, concept, or thing about which data is stored in a database. It could be a person, place, thing, event, or concept that can be uniquely identified and distinguished from other entities. In an Entity-Relationship (ER) diagram, entities are represented by rectangles.

Attributes: Attributes are the characteristics or properties that describe an entity. They represent the specific pieces of information associated with an entity. Attributes can have different types, such as:

Simple Attribute: An attribute that cannot be divided into smaller parts. For example, "Name" or "Age".

Composite Attribute: An attribute that can be divided into smaller sub-parts, each representing a simpler attribute. For example, "Address" composed of "Street", "City", "State", and "Zip Code".

Derived Attribute: An attribute whose value is derived from other attributes. For example, "Age" can be derived from the "Date of Birth".

Multi-valued Attribute: An attribute that can hold multiple values for a single entity. For example, "Phone Numbers" for a person.

Key Attribute: An attribute whose values uniquely identify an entity within an entity set. For

One-to-Many (1:M) Relationship: Each entity in the first entity set can be associated with many entities in the second entity set, but each entity in the second entity set is associated with only one entity in the first entity set.

Many-to-Many (M:N) Relationship: Entities in both entity sets can be associated with many entities in the other entity set. This type of relationship requires the use of an associative entity (also known as a junction or link entity) to represent the association between them.