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:**

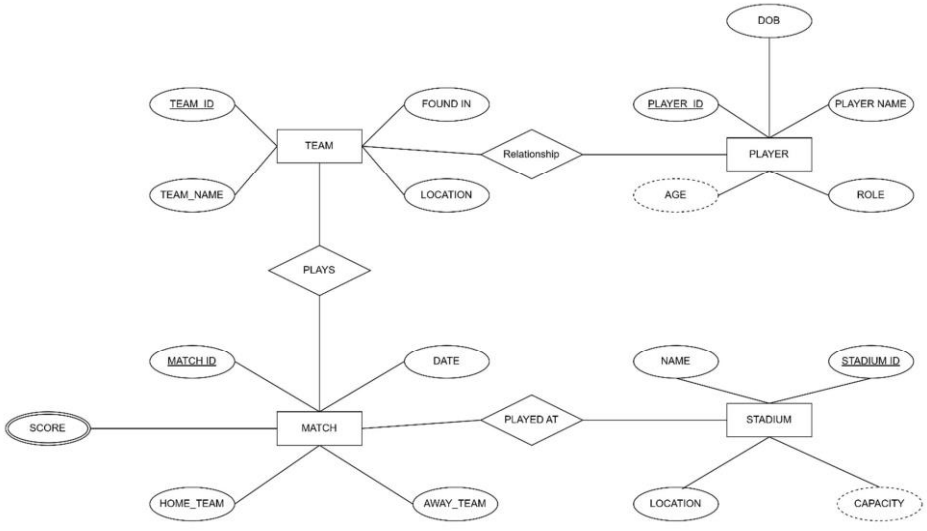
- Relationships describe how entities are related to each other or how they interact.
- They represent the associations between entities.
- Relationships are depicted as lines connecting related entities in the ER diagram.
- Each relationship has a degree, indicating the number of entities involved. 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.

Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

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

Entity: Represents a real-world object or concept in a database.
Attributes: Characteristics of entities with defined data types.
Relationships: Connections between entities, defining how they interact or associate with each other.

2. Write ER/EER diagram notations

Entities: Rectangles or ovals.
Attributes: Ellipses connected to entities by lines.
Primary Key Attribute: Underlined attribute.
Composite Attribute: Attributes nested within ellipses.
Multivalued Attribute: Double oval.
Derived Attribute: Dashed ellipse.
Relationships: Diamonds or lines connecting entities.
Cardinality: Indicates the minimum and maximum number of instances in a relationship using crow's foot notation.
Participation Constraint: Total participation denoted by a double line.