AIM:	ER Modeling
Program 1	
PROBLEM STATEMENT:	Draw the ER Model Diagram for the given case study system.
Theory:	1. Data Model: Collection of conceptual tools for describing data, data relationships, datasemantics, and consistency constraints. 2. ER Model: 1. It is a high level data model based on a perception of a real world that consists of acollection of basic objects, called entities and of relationships among these objects. 2. Graphical representation of ER Model is ER diagram, which acts as a blueprint ofDB. 3. Entity: An Entity is a “thing” or “object” in the real world that is distinguishable from all otherobjects. 1. It has physical existence. 2. Each student in a college is an entity. 3. Entity can be uniquely identified. (By a primary attribute, aka Primary Key) 4. Strong Entity: Can be uniquely identified. 5. Weak Entity: Can’t be uniquely identified., depends on some other strong entity. 1. It doesn’t have sufficient attributes, to select a uniquely identifiable attribute. 2. Loan -> Strong Entity, Payment -> Weak, as instalments are sequential number countercan be generated separate for each loan. 3. Weak entity depends on strong entity for existence.

4. Entity set:

1. It is a set of entities of the same type that share the same properties, or attributes.
2. E.g., Student is an entity set.
3. E.g., Customer of a bank

5. Attributes:

1. An entity is represented by a set of attributes.
2. Each entity has a value for each of its attributes.
3. For each attribute, there is a set of permitted values, called the domain, or value set, of that attribute.
4. E.g., Student Entity has following attributes
    A. Student_ID
    B. Name
    C. Standard
    D. Course
    E. Batch
    F. Contact number
    G. Address

Types of Attributes:

1. Simple

1. Attributes which can't be divided further.
2. E.g., Customer's account number in a bank, Student's Roll number etc.

2. Composite

1. Can be divided into subparts (that is, other attributes).
2. E.g., Name of a person, can be divided into first-name, middle-name, last-name.
3. If user wants to refer to an entire attribute or to only a component of the attribute.
4. Address can also be divided, street, city, state, PIN code.

3. Single-valued

1. Only one value attribute.
2. e.g., Student ID, loan-number for a loan.

4. Multi-valued

1. Attribute having more than one value.
2. e.g., phone-number, nominee-name on some insurance, dependent-name etc.
3. Limit constraint may be applied, upper or lower limits.

5. Derived

1. Value of this type of attribute can be derived from the value of other related attributes.
2. e.g., Age, loan-age, membership-period etc.

6. Relationships:

1. Association among two or more entities.
2. e.g., Person has vehicle, Parent has Child, Customer borrow loan etc.
3. Strong Relationship, between two independent entities.
4. Weak Relationship, between weak entity and its owner/strong entity.
1. e.g., Loan <instalment-payments> Payment.

5. Degree of Relationship

1. Number of entities participating in a relationship.
2. ***Unary***, Only one entity participates. e.g., Employee manages employee.
3. ***Binary***, two entities participates. e.g., Student takes Course.
4. ***Ternary*** relationship, three entities participates. E.g, Employee works-on branch, employee works-on job.
5. Binary are common.

7. Relationships Constraints:

1. Mapping Cardinality / Cardinality Ratio
1. Number of entities to which another entity can be associated via a relationship.
2. **One to one**, Entity in A associates with at most one entity in B, where A & B are entity sets. And an entity
of B is associated with at most one entity of A.
1. E.g., Citizen has Aadhar Card.
3. **One to many**, Entity in A associated with N entity in B. While entity in B is associated with at most one
entity in A.
1. e.g., Citizen has Vehicle.
4. **Many to one**, Entity in A associated with at most one entity in B. While entity in B can be associated with
N entity in A.
1. e.g., Course taken by Professor.
5. **Many to many**, Entity in A associated with N entity in B. While entity in B also associated with N entity in A.
1. Customer buys product.
2. Student attend course.

Extended features of ER Model :

1. Basically ER Features can be used to model most DB features but when complexity increases, it is better to use some Extended ER features to model the DB Schema.

2. Specialisation

1. In ER model, we may require to subgroup an entity set into other entity sets that are distinct in some way with other entity sets.
2. Specialisation is splitting up the entity set into further sub entity sets on the basis of their functionalities, specialities and features.
3. It is a Top-Down approach.
4. e.g., Person entity set can be divided into customer, student, employee. Person is superclass and other specialised entity sets are subclasses. 1. We have “is-a” relationship between superclass and subclass. 2. Depicted by triangle component. 5. Why Specialisation? 1. Certain attributes may only be applicable to a few entities of the parent entity set. 2. DB designer can show the distinctive features of the sub entities. 3. To group such entities we apply Specialisation, to overall refine the DB blueprint.

3. Generalisation

1. It is just a reverse of Specialisation. 2. DB Designer, may encounter certain properties of two entities are overlapping. Designer may consider to make a new generalised entity set. That generalised entity set will be a super class. 3. “is-a” relationship is present between subclass and super class. 4. e.g., Car, Jeep and Bus all have some common attributes, to avoid data repetition for the common attributes. DB designer may consider to Generalise to a new entity set “Vehicle”. 5. It is a Bottom-up approach. 6. Why Generalisation? 1. Makes DB more refined and simpler. 2. Common attributes are not repeated.

4. Attribute Inheritance

1. Both Specialisation and Generalisation, has attribute inheritance.
2. The attributes of higher level entity sets are inherited by lower level entity sets.
3. E.g., Customer & Employee inherit the attributes of Person.

5. Participation Inheritance

1. If a parent entity set participates in a relationship then its child entity sets will also participate in that relationship.

6. Aggregation

1. How to show relationships among relationships? - Aggregation is the technique.

2. Abstraction is applied to treat relationships as higher-level entities. We can call it Abstract entity.
Avoid redundancy by aggregating relationship as an entity set itself.

Symbols used in ER Model Diagram :

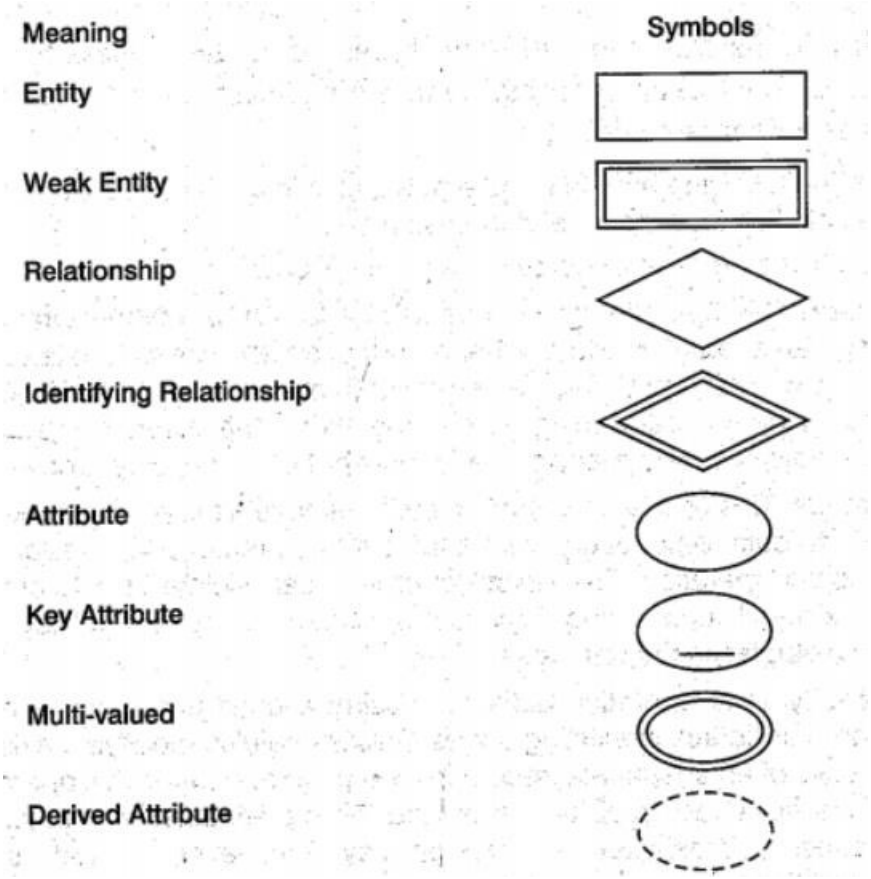

Fig no. 1.1 : Symbols(Notations) used to draw ER model diagram

Queries

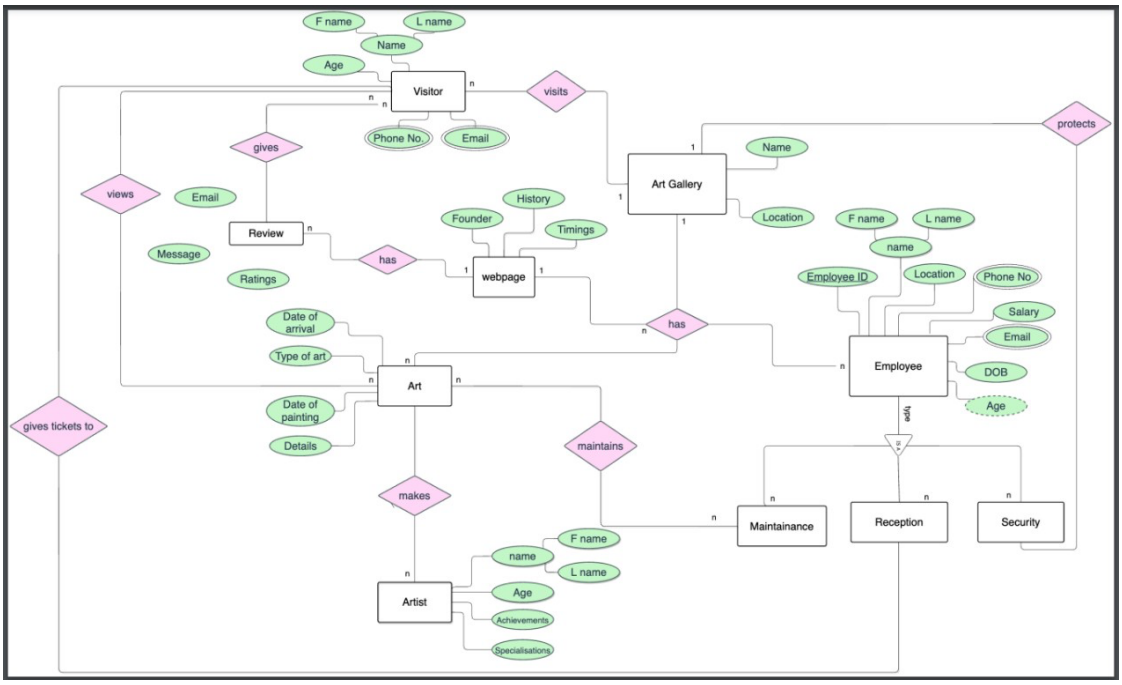

Fig no. 1.2 : ER Model Diagram of Art gallery

Conclusion:

1. From this experiment I learned that, In the database when the data stored in the form of tables, tables are connected to each other by making relationships in between them. The entities of system, attributes of entities, relationship, cardinality is graphically represented in the form of ER model.
2. Now I'm known to how to create relationship between the entities, making ER model diagram on given case study ,defining the attributes of weak or strong entity perfectly.
3. Also, I'm successfully present our ER model diagram in front of my friends.