LAB SESSION 3

TITLE: ENTITY RELATIONSHIP DIAGRAM

DATE:

Problem Definition:
Draw the Entity Relationship diagram for the project using ERD software.

Software Used:
LUCID CHART
DRAW.IO

Theory:
Entity-Relationship Diagrams (ERDs) are fundamental tools in database modeling and design. They provide a graphical representation of entities and their relationships within a database system, allowing designers and stakeholders to visualize complex data structures and interaction.

Entities

- **Definition and Representation:**
  Entities represent real-world objects or concepts with a distinct existence in the database. In ERDs, entities are typically represented by rectangles containing the entity name.
- **Examples:**
  - Physical objects: Customer, Product, Building
  - Concepts: Order, Invoice, Reservation

Types of Entities

- **Strong Entity:**
  A strong entity can be uniquely identified by its own attributes.
  - Characteristics:
    - Has a primary key
    - Exists independently of other entities
  - Example: A Customer entity with a unique customer ID as its primary key
- **Weak Entity:**
  A weak entity cannot be uniquely identified by its own attributes alone and depends on a strong entity (owner entity) for its existence.
  - Characteristics:
    - Has a partial key (discriminator)
    - Requires the primary key of its owner entity for unique identification
  - Example: An OrderItem entity that depends on the Order entity for its existence and unique identification

Attributes

- **Definition and Representation:**
  Attributes are properties or characteristics of an entity. In ERDs, attributes are typically represented by ovals connected to their entity by a line.

Types of Attributes

- **Single-valued Attributes:**
  Hold a single value for a particular entity instance.
  - Example: A person's age

- **Multi-valued Attributes:**
  Can hold multiple values for a particular entity instance.
  - Example: A person's phone numbers
- **Composite Attributes:**
  Can be divided into smaller sub-parts, each with independent meanings.
  - Example: A person's address
- **Derived Attributes:**
  Can be derived from other attributes or entities.
  - Example: A person's age derived from their date of birth
- **Stored Attributes:**
  Actual stored data values in the database.
  - Example: A person's name directly stored in the database

Relationships

- **Definition and Representation:**
  Relationships define how entities interact with each other. In ERDs, relationships are typically represented by diamonds connected to related entities by lines.

Types of Relationships

- **One-to-One (1:1)**
  A single instance of one entity relates to a single instance of another entity.
  - Example: One person has one passport number
- **One-to-Many (1:M)**
  A single instance of one entity relates to multiple instances of another entity.
  - Example: One department has many employees
- **Many-to-One (M:1)**
  Multiple instances of one entity relate to a single instance of another entity.
  - Example: Many employees work in one department
- **Many-to-Many (M:N)**
  Multiple instances of one entity relate to multiple instances of another entity.
  - Example: Students can enrol in multiple courses, and courses can have multiple students

Participation

- **Total Participation:**
  Every instance of an entity must participate in at least one instance of the relationship.
  - Example: Every employee must belong to a department
- **Partial Participation:**
  Some instances of an entity may not participate in any instance of the relationship.
  - Example: Not every person needs to own a car

Cardinality

Cardinality specifies the number of instances of one entity that can or must be associated with each instance of another entity.

- **Examples of cardinality notation:**
  - Zero or one
  - Exactly one
  - Zero or many
  - One or many

Keys
Keys are attributes or sets of attributes that uniquely identify an entity within its entity set.

Types of Keys
- **Primary Key:**
  Unique identifier for an entity.
  - Example: Customer ID for a Customer entity
- **Candidate Key:**
  Attribute(s) that can uniquely identify a tuple in a relation.
  - Example: Both SSN and employee ID could be candidate keys for an Employee entity
- **Super Key:**
  Set of attributes that collectively identify a tuple uniquely.
  - Example: SSN and Name for an Employee entity
- **Foreign Key:**
  An attribute in one table that is a primary key in another table.
  - Example: Department ID in an Employee table referencing the Department table
- **Composite Key:**
  Consists of two or more attributes that together uniquely identify an entity.
  - Example: Order ID and Product ID might form a composite key for an OrderItem entity
- **Alternate Key:**
  A candidate key not chosen as the primary key.
  - Example: If employee ID is the primary key, SSN could be an alternate key
- **Surrogate Key:**
  System-generated unique identifier not derived from application data.
  - Example: An auto-incremented ID assigned by the database system

Advanced Concepts
- **Weak Entities and Identifying Relationships:**
  Weak entities depend on strong entities and are connected by identifying relationships.
  - Example: The primary key of a weak entity includes the primary key of its owner entity
- **Associative Entities (Bridge Entities):**
  Used to break down many-to-many relationships into two one-to-many relationships.
  - Example: Associative entities often contain additional attributes specific to the relationship
- **Generalization and Specialization:**
  Represents inheritance hierarchies.
  - Example: Generalization combines lower-level entities into a higher-level entity, while Specialization divides a higher-level entity into lower-level entities
- **Aggregation:**
  Treats a relationship itself as an entity.
  - Example: This whole structure can then participate in a relationship with another entity
- **Recursive Relationships:**
  An entity is related to itself.
  - Example: An Employee entity with a "manages" relationship to itself

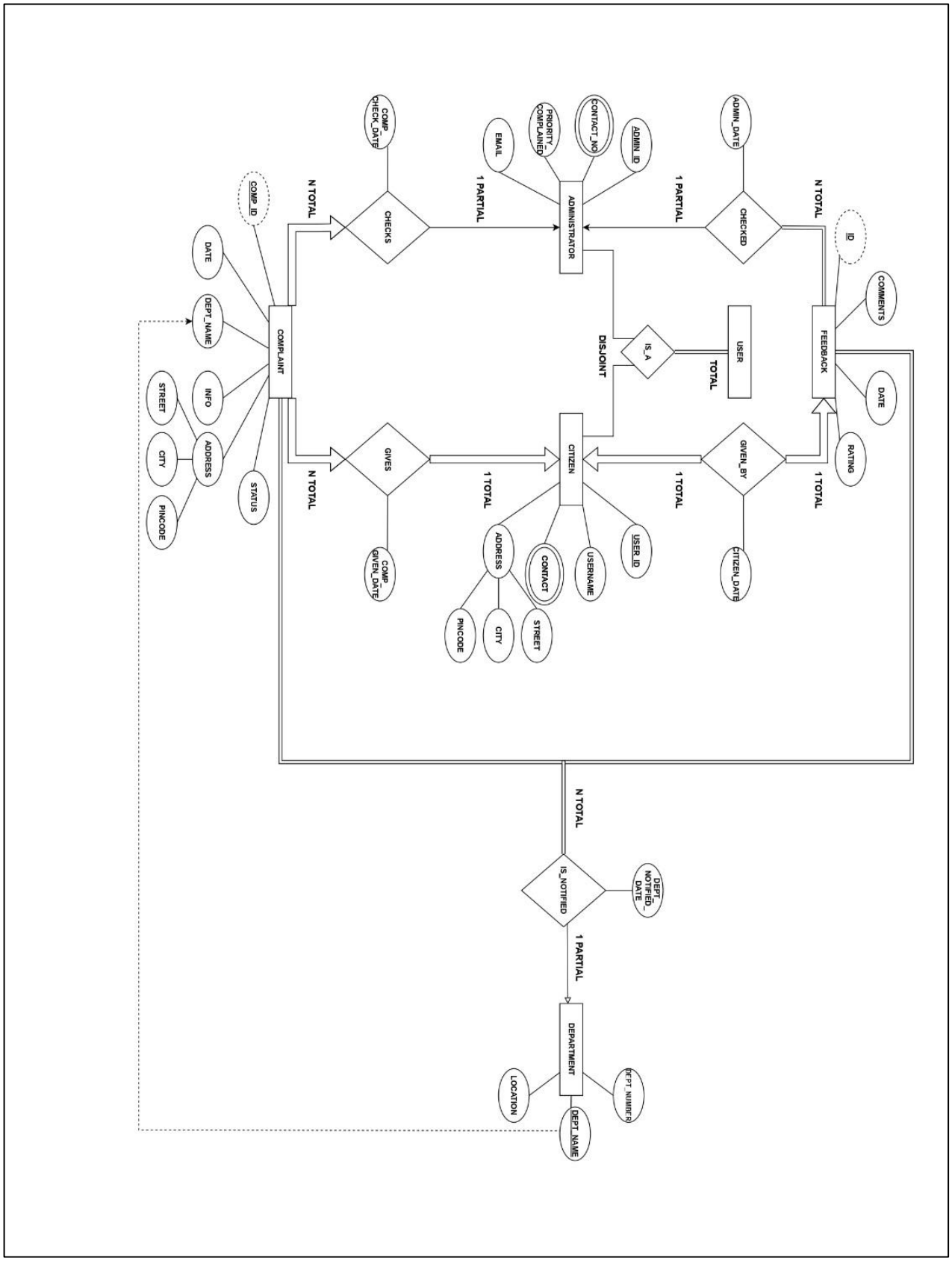
- **Identifying Relationships:**
  A relationship between a weak entity and its owner (strong) entity.
  - Example: In an Order (strong entity) and OrderItem (weak entity) relationship, each OrderItem cannot exist without its associated Order

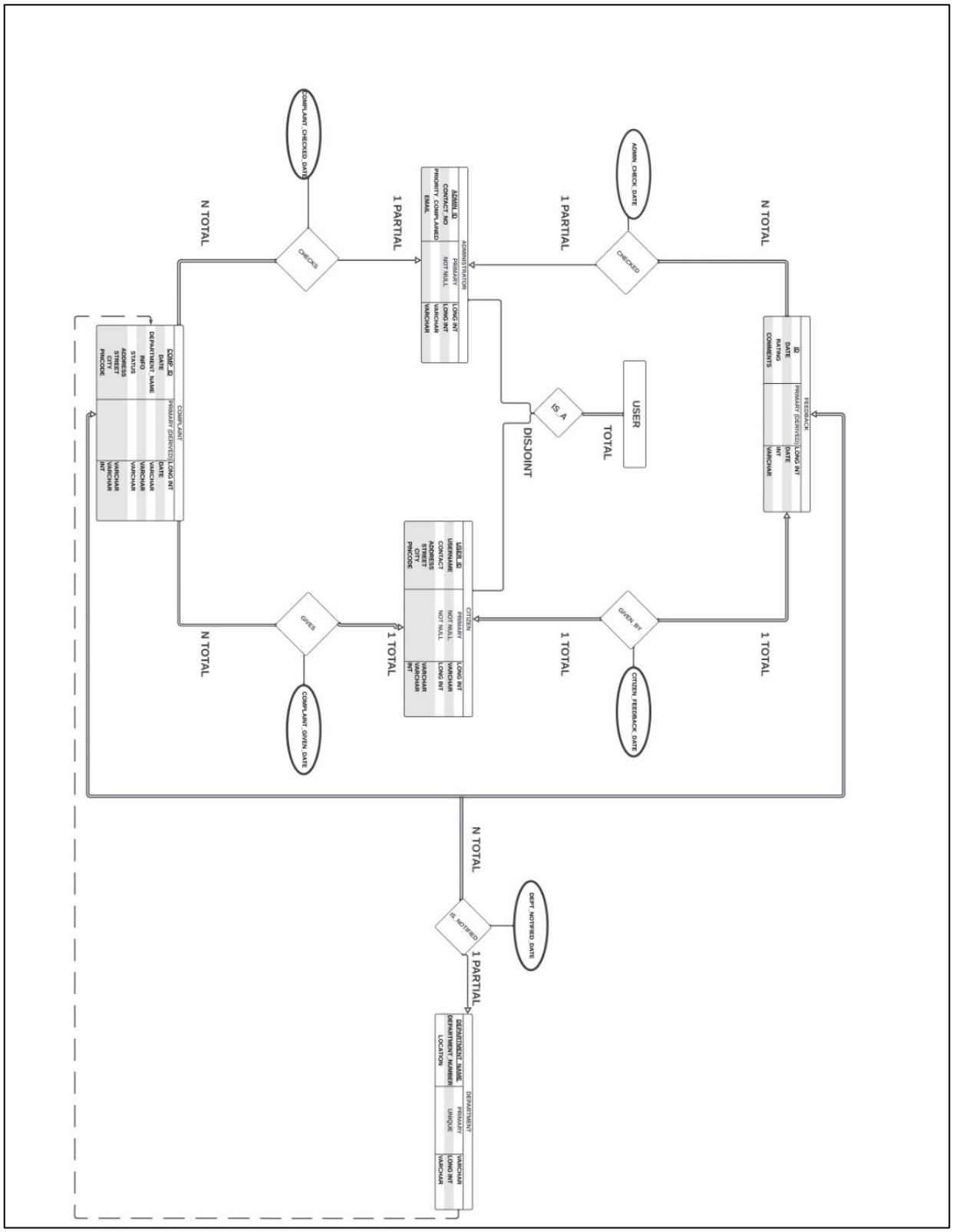
- **Non-Identifying Relationships:**
  Relationships where the child entity can exist independently of the parent entity.
  - Example: A Department and Employee relationship where an Employee can exist even if their Department is deleted
- **Discriminator:**
  An attribute of the supertype entity that determines which subtype entity each instance belongs to.
  - Example: VehicleType could be a discriminator attribute in the Vehicle entity

Degree of Relationships
The degree of a relationship refers to the number of entities participating in the relationship.

- **Binary Relationships:**
  A relationship between two entities.
  - Example: An Employee works in a Department (Many-to-One)
- **Ternary Relationships:**
  A relationship involving three entities simultaneously.
  - Example: A Student enrolls in a Course taught by an Instructor
- **N-ary Relationships:**
  Relationships involving more than three entities.

Output/Results





Conclusion:

Entity-Relationship Diagrams provide a structured way to visualize the data model of a system, making it easier to understand and communicate the database structure and relationships. By using these components and concepts, we can create comprehensive and accurate representations of complex data systems.