Table of Contents

List of Figure

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to everyone who has contributed to the successful completion of this Entity-Relationship (ER) diagram for voting online site.

First and foremost, I would like to extend my heartfelt gratitude to my esteemed lecturer, Mr. Manish Rathi, for his invaluable guidance, unwavering support, and profound expertise throughout the project. His insightful perspectives and constructive feedback have been instrumental in helping me create a comprehensive and effective ER diagram.

I am also deeply indebted to my esteemed classmates, who have been an unwavering source of encouragement, motivation, and constructive critique during the course of this project. Their invaluable contributions have played a pivotal role in refining and enhancing the ER diagram, and I am truly grateful for their willingness to collaborate and share their ideas. Working alongside such a talented and dedicated group of individuals has been an enriching experience, and I have gained immense knowledge from their diverse perspectives and experiences.

I am profoundly grateful for the contributions of all those who have supported me in this endeavor, and I sincerely hope that this ER diagram will contribute to the development of a more efficient and user-friendly voting online site.

ABSTRACT

The Entity-Relationship (ER) diagram serves as a fundamental tool in database design, providing a visual representation of the logical structure and interrelationships among various entities within a system. This report presents a comprehensive exploration of ER diagrams, detailing their significance, components, and practical applications.

Beginning with an introduction to ER diagrams, the report elucidates their role as a conceptual blueprint for designing databases, outlining the fundamental components and their relationships within the database structure. It highlights the importance of ER diagrams in providing a clear and visual representation of data relationships, aiding in understanding, documentation, and maintenance of the database structure.

A detailed examination of the symbols used in ER diagrams is provided, offering insights into the representation of entities, attributes, and relationships. Through illustrative examples and figures, the report elucidates the visual depiction of strong and weak entities, key attributes, composite attributes, multivalued attributes, and derived attributes, facilitating a deeper understanding of ER diagram components.

Moreover, the report delves into the various types of relationships depicted in ER diagrams, including one-to-one, one-to-many, many-to-one, and many-to-many relationships. Each type is elucidated with clear examples, elucidating the manner in which entities interact and establish associations within a database system.

Furthermore, the report outlines the step-by-step process for creating an ER diagram, emphasizing the importance of proper entity identification, relationship determination, attribute attachment, and removal of redundant elements. Additionally, it underscores the significance of color coding to enhance the visualization and comprehension of database structures.

In conclusion, the report presents an ER diagram of an online voting site as a practical application, showcasing the implementation of the concepts elucidated throughout the document. Through this exploration, readers gain a comprehensive understanding of ER diagrams and their pivotal role in database design and management.

1. INTRODUCTION TO ENTITY RELATIONSHIP (ER)-DIAGRAM

An Entity-Relationship (ER) diagram is a visual representation used in database design to depict the logical structure and relationships among various entities within a system. It's a powerful tool for modeling the essential components of a database and how they relate to one another.

An Entity-Relationship (ER) model serves as a conceptual representation or blueprint for designing a database. It outlines the fundamental components and their relationships within the database structure.

1.1. DEFINITION

An Entity-Relationship (ER) diagram visually represents a data model, showcasing the entities, their attributes, and the relationships between entities within a system or database. Creating an ER diagram is a structured approach to database design, necessitating a thorough analysis of all data requirements prior to implementation.

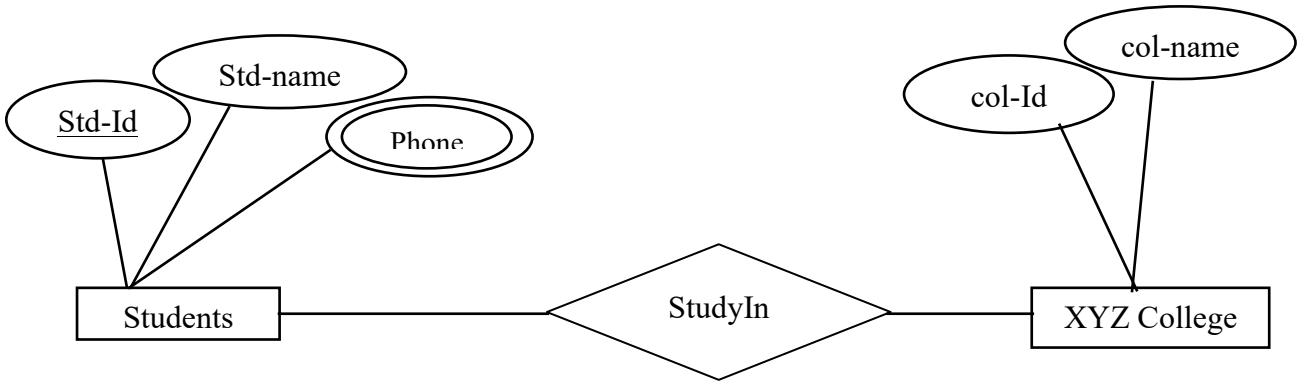


Fig 1.1: ER Diagram Introduction

1.2. SIGNIFICANCE

- ER diagrams provide a clear and visual representation of how different pieces of data relate to each other in a database.
- ER diagrams depict how different entities are linked, making it easy to understand how data is related and used.
- ER diagrams act as documentation of the database structure, aiding future maintenance and updates.
- ER diagrams help ensure that the database design meets the requirements of the system it supports.

1.3. SYMBOLS USED IN ER DIAGRAM

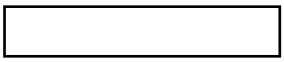
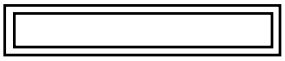
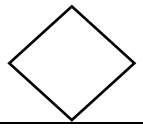
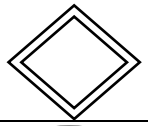
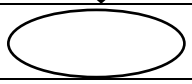
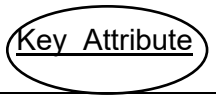
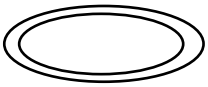
ER COMPONENT	DESCRIPTION	SYMBOLS
Strong Entity	Rectangle box	
Weak Entity	Double Rectangle	
Relationship	Rhombus	
Weak entity	Double lined rhombus	
Attributes	Ellipse	
Key Attribute	Underline the attribute name inside Ellipse	
Derived Attribute	Dotted Ellipse	
Multivalued Attribute	Double ellipse	

Fig 1.3.: Symbol used in ER-Diagram

2. COMPONENT OF ER DIAGRAM

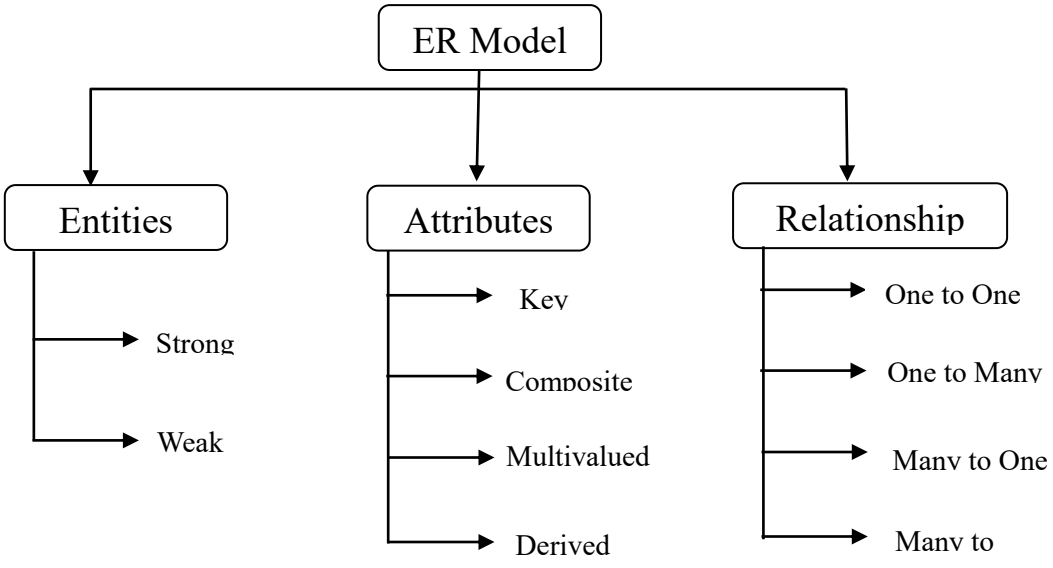


Fig 2: Component of ER Diagram

2.1. ENTITY

An entity is a fundamental component in an ER diagram, representing a distinct object, concept, or thing in the real world that is relevant to the system being modeled. It can have attributes (properties or characteristics) that describe it. Entities is represented as a rectangle with rounded corner.

2.1.1. Strong Entity:

A strong entity is an entity set that possesses a key attribute or a combination of attributes that uniquely identifies each instance within the set. It corresponds to a table in a database that contains a primary key column. Strong entities are depicted by a single rectangular shape in an Entity-Relationship (ER) diagram.

When representing the association between two strong entities, a single diamond symbol is utilized. This diamond symbol connects the two rectangular shapes representing the strong entities through lines, effectively illustrating the relationship that exists between them.

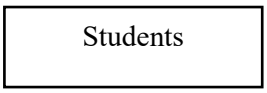


Fig 2.1.1: Symbol of Strong Entity

2.1.2. Weak Entity:

A weak entity set is an entity set that does not possess a primary key attribute or a combination of attributes that can uniquely identify its individual entities. Unlike strong entity sets, which have a self-contained primary key, weak entity sets rely on the primary key of a related strong entity set for unique identification. However, weak entity sets do contain a partial key, known as a discriminator, which can identify groups of entities within the set but cannot distinguish between individual entities.

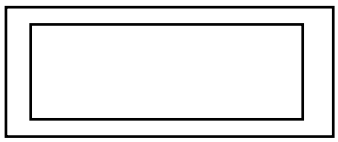


Fig 2.1.2: Symbol of weak Entity

2.2. ATTRIBUTES

An attributes describe the property of an entity. It is represented as Oval in ER Diagram. Types are described below: -

2.2.1. Key Attributes

A key attribute possesses the ability to uniquely identify each individual entity within an entity set. In an Entity-Relationship (ER) diagram, the visual representation of a key attribute is achieved by underlining the text corresponding to that specific attribute. For instance, “Employee ID” could be a key attribute for the “Employee” entity.

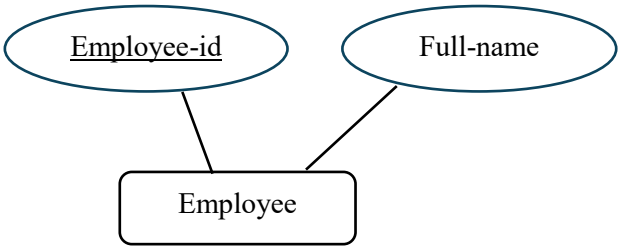


Fig 2.2.1: Key Attributes

2.2.2. Composite Attributes

These attributes can be divided into smaller sub-parts, representing more basic attributes with independent meanings. For instance, a “Date” attribute can be divided into “year” and “month”.

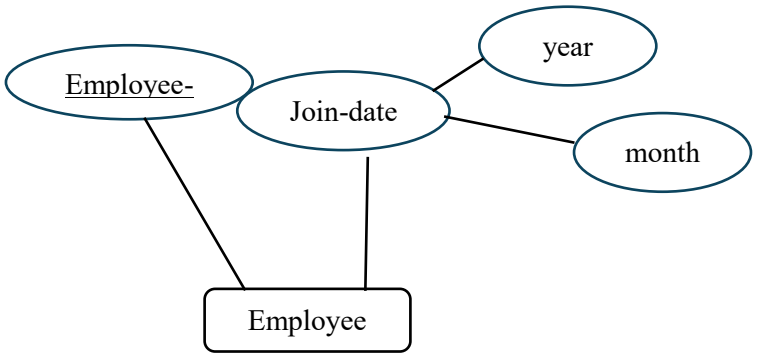


Fig 2.2.2: Composite Attributes

2.2.3. Multivalued Attributes

Some attributes can possess over one value, those attributes are called multivalued attributes. The double oval shape is used to represent a multivalued attribute. For instance, a "language spoken" attribute might be multi-valued if employees can have more than one phone number.

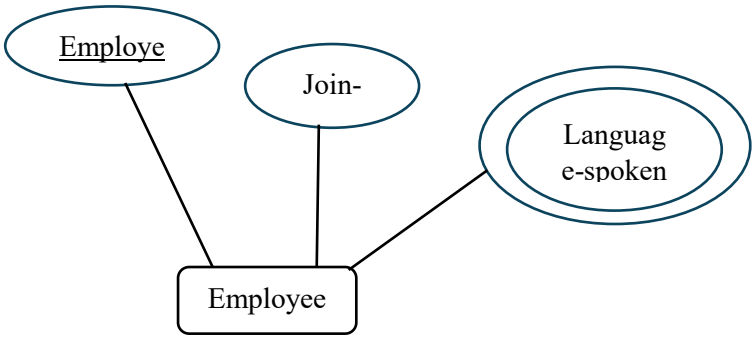


Fig 2.2.3: Multivalued Attributes

2.2.4. Derived Attributes: A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by dashed oval in an ER Diagram. For example, an employee's years can be derived from their "Join Date".

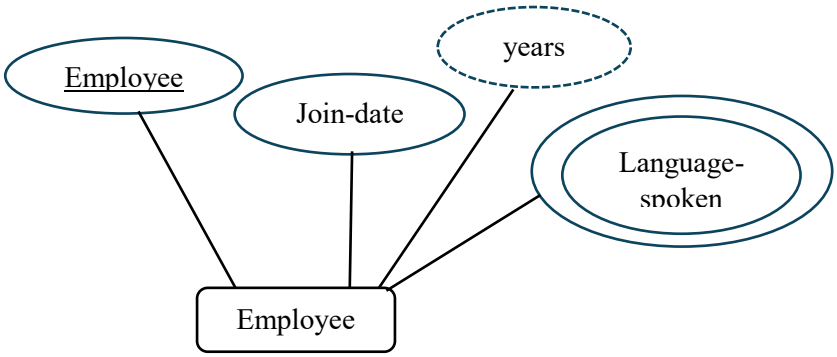


Fig 2.2.4: Derived Attributes

3. Relationship

A Relationship is a logical link between two or more entities, depicting how the entities interact with each other. Relationships help to define the associations and dependencies between entities in a database.

There are four types of relationship: -

3.2. One to One Relationship: When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship.

For examples, a citizen has only one citizenship card and an identification card is given to one person.

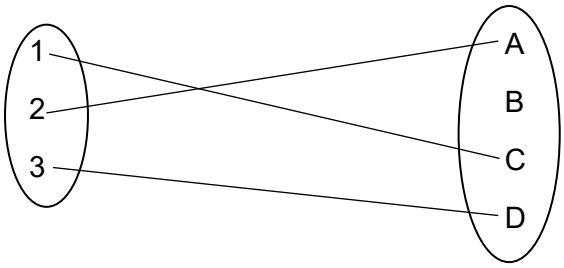


Fig 3.1.1. : One to One Relationship (Mapping)

Fig 3.1.2.: Notation used in ER Diagram for One to One Relationship

3.3. One to Many Relationship

This is a relationship where a single instance of an entity A can be associated with multiple instances of entity B, but each instance of entity B can be associated with only one instance of entity A. For example, a 'Department' entity can have a one-to-many relationship with an 'Employee' entity, representing that one department can have many employees, but each employee belongs to one department.

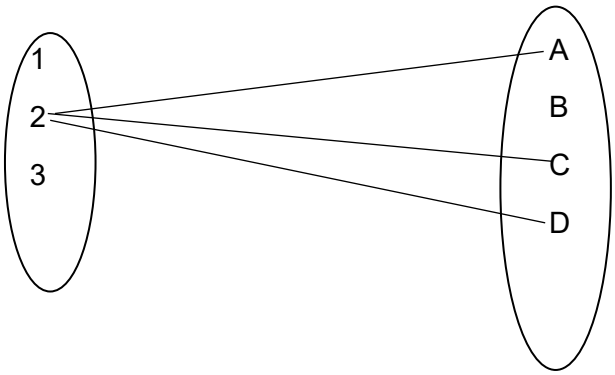


Fig 3.2.1.: One to Many Relationship (Mapping)

Fig 3.2.2.: Notation used in ER Diagram for One to Many Relationship

3.4. Many to One Relationship

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as many to one relationship. For Example many student can enroll one same course.

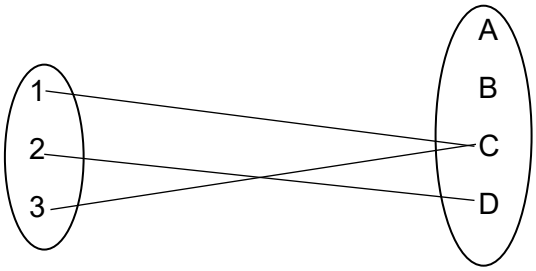


Fig 3.3.1.: Many to One Relationship (Mapping)

Fig 3.3.2.: Notation used in ER Diagram for Many to One Relationship

3.5. Many to Many Relationship

When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, an 'Author' entity can have a many-to-many relationship with a 'Book' entity, indicating that one author can write many books, and one book can have many authors.

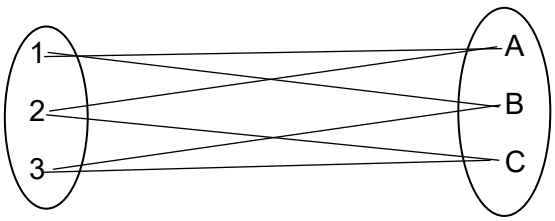


Fig 3.4.1.: Many to Many Relationship (Mapping)

Fig 3.4.2.: Notation used in ER Diagram for Many to Many Relationship

4. How to Draw ER Diagram?

The process to draw ER Diagram is given steps followed: -

- The very first step is identifying all the Entities, and place them in a Rectangle, and labeling them accordingly.
- The next step is to identify the relationship between them and pace them accordingly using the Diamond, and make sure that, Relationships are not connected to each other.
- Attach attributes to the entities properly.
- Remove redundant entities and relationships.
- Add proper colors to highlight the data present in the database.

5. ER Diagram of Online Voting Site

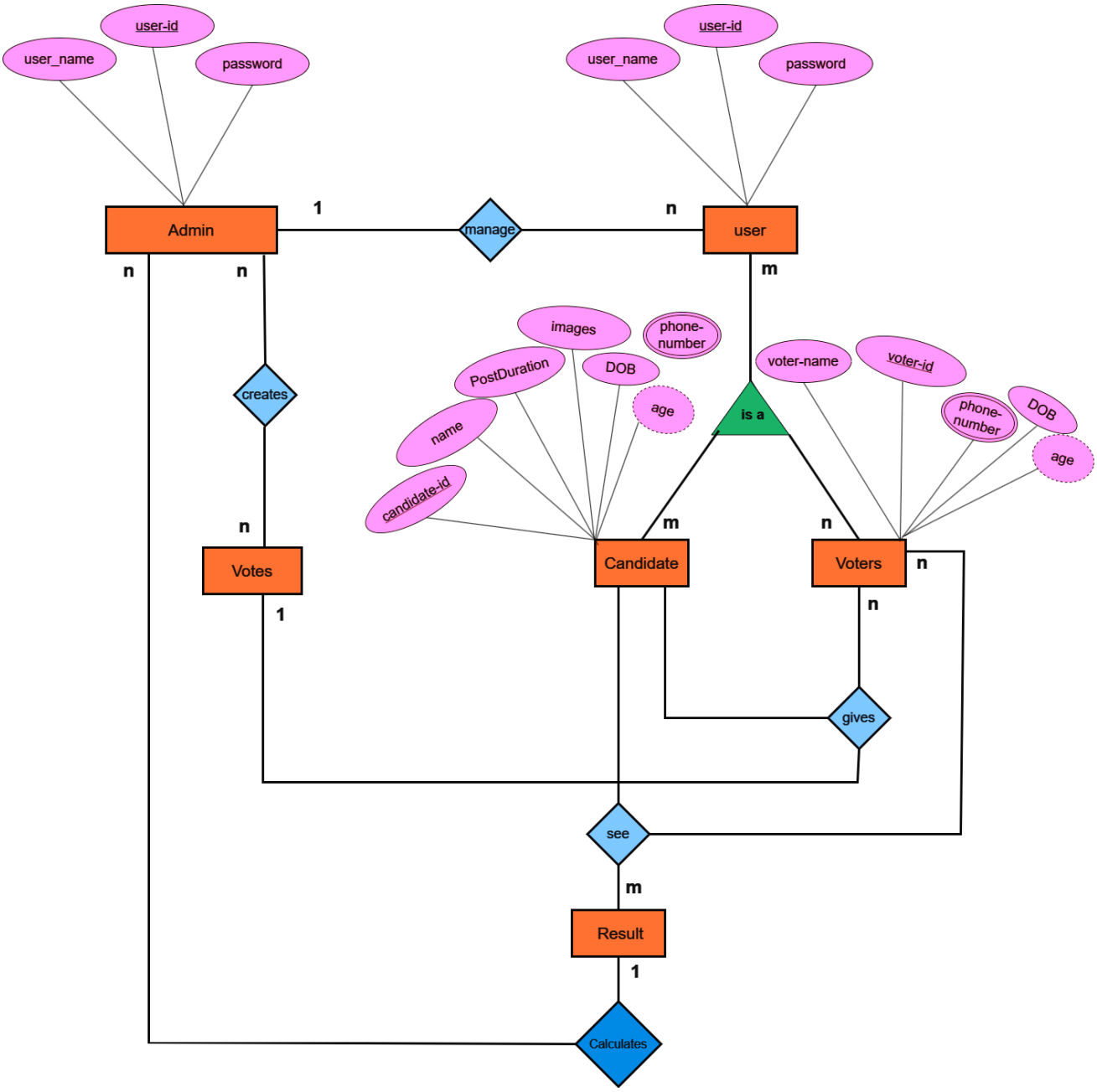


Fig 5: ER Diagram of Online Voting site

The ER diagram provides a detailed look at a complex voting system, illustrating the relationships between six main components: Admin, User, Candidate, Voter, Vote, and Result. Admins, acting as overseers, have unique identifiers (user_id) and access credentials (user_name, password) to manage Users. Users, inheriting from a central entity, can take on roles as either Candidates or Voters, simplifying user management.

Candidates, identifiable through attributes like candidate_id and name, actively engage in the voting process. They interact with Voters in a dynamic many-to-many relationship, allowing for extensive engagement with different Voters, fostering comprehensive participation.

Voters, similar to Candidates, inherit basic User traits and serve as the recipients of votes. They partake in the election process by receiving votes from multiple Candidates, influencing the outcome through their decisions.

The Votes entity, initiated by Admins, serves as the conduit for conducting elections. Admins oversee the creation and distribution of Votes among Candidates and Voters, ensuring the fairness and transparency of the voting process.

The Result entity, derived from Votes, represents the conclusion of the voting process. It establishes a many-to-one relationship with Candidates, permitting multiple Candidates to access and interpret the election outcome.

In summary, the ER diagram demonstrates a well-structured voting system where Admins manage Users, who can take on roles as either Candidates or Voters. Candidates engage with Voters through dynamic interactions facilitated by the many-to-many relationship. The creation of Votes by Admins and the computation of Results ensure an efficient and transparent election process, promoting trust and credibility within the system.

6. CONCLUSION

This report explores the importance and construction of Entity-Relationship (ER) diagrams, particularly for an online voting system. ER diagrams are vital in database design, providing a clear visual representation of data interactions and relationships.

We began with an introduction to ER diagrams, outlining their purpose and symbols. The report detailed components such as strong and weak entities, various attributes, and entity relationships, accompanied by a step-by-step guide for creating accurate ER diagrams.

A practical example was provided through an ER diagram for an online voting system, showcasing interactions between entities like Admin, User, Candidate, Voters, Votes, and Result. This example mapped out the system's structure, ensuring a comprehensive understanding of its database.

In summary, ER diagrams are essential for designing efficient databases. They visualize data relationships, facilitating easier development, documentation, and maintenance. This report underscores both the theoretical and practical aspects of ER diagrams, highlighting their role in creating effective database designs.