Experiment No.2

Mapping ER/EER to Relational schema model.

Aim :- Prepare the schema for Relational Model with the ER/ERR diagram, drawn for the identified case study in experiment no.1.

Objective :- To map the Entity Relationship (ER) / Extended Entity-Relationship (EER) Diagram to Relational Model schema and learn to incorporate various schema-based constraints.

Theory:

Mapping an Entity-Relationship (ER) model to a relational database schema involves translating the conceptual model represented in the ER diagram into tables and relationships in a relational database management system (DBMS). Here are the general rules for mapping ER to a schema in a DBMS:

1. Entities to Tables:
    a. Each entity in the ER diagram corresponds to a table in the relational schema.
    b. The attributes of the entity become the columns of the table.
    c. The primary key of the entity becomes the primary key of the table.

2. Relationships to Tables:
    a. Many-to-Many Relationships:
        i. Convert each many-to-many relationship into a new table.
        ii. Include foreign key columns in this table to reference the participating entities.
        iii. The primary key of this table may consist of a combination of the foreign keys from the participating entities.
    b. One-to-Many and One-to-One Relationships:
        i. Represented by foreign key columns in one of the participating tables.
        ii. The table on the "many" side of the relationship includes the foreign key column referencing the table on the "one" side.
        iii. The foreign key column typically references the primary key of the related table.

3. Attributes to Columns:
    a. Each attribute of an entity becomes a column in the corresponding table.

   b. Choose appropriate data types for each attribute based on its domain and constraints.
   c. Ensure that attributes participating in relationships are represented as foreign keys when needed.

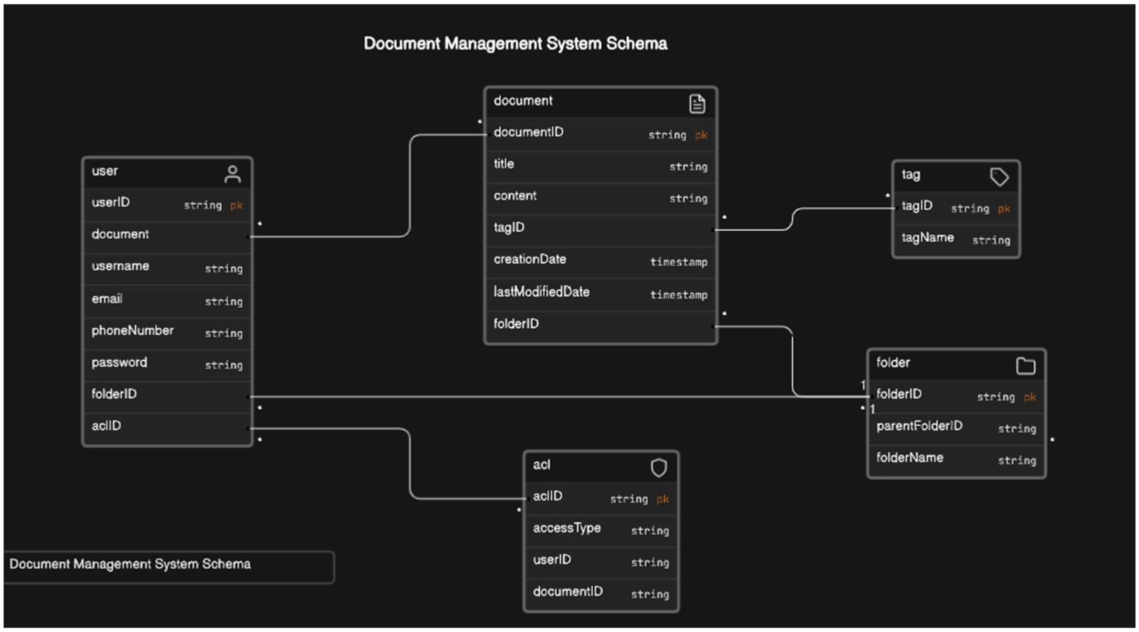
4. Primary and Foreign Keys:
   a. Identify the primary key(s) of each table based on the primary key(s) of the corresponding entity.
   b. Ensure referential integrity by defining foreign keys in tables to establish relationships between them.
   c. Foreign keys should reference the primary key(s) of related tables.
   d. Ensure that foreign keys have appropriate constraints, such as ON DELETE CASCADE or ON UPDATE CASCADE, to maintain data integrity.

5. Cardinality Constraints:
   a. Use the cardinality constraints from the ER diagram to determine the multiplicity of relationships in the relational schema.
   b. Ensure that the constraints are enforced through the appropriate use of primary and foreign keys.

6. Normalization:
   a. Normalize the schema to minimize redundancy and dependency.
   b. Follow normalization rules such as First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), etc., to ensure data integrity and minimize anomalies.

7. Indexing and Optimization:
   a. Consider indexing frequently queried columns to improve query performance.
   b. Evaluate the schema design for optimization opportunities based on query patterns and performance requirements.

Implementation:



Document Management System Schema

user	
userID	string pk
document	
username	string
email	string
phoneNumber	string
password	string
folderID	
aclID	

document	
documentID	string pk
title	string
content	string
tagID	
creationDate	timestamp
lastModifiedDate	timestamp
folderID	

tag	
tagID	string pk
tagName	string

folder	
folderID	string pk
parentFolderID	string
folderName	string

acl	
aclID	string pk
accessType	string
userID	string
documentID	string

Document Management System Schema

Conclusion:

1. write definition of relational schema and notations

➔ **Relational Schema**:

- A **relational schema** is a logical representation of how data should be stored in a relational database.
- It defines the structure of tables (relations), their attributes (columns), and the relationships between them.
- A schema provides a blueprint for organizing data and ensures data integrity.
- **Notations**:
  - In ER diagrams, we use the following symbols to represent relational schema components:
    - **Rectangles**: Represent entities (tables).
    - **Ellipses (Ovals)**: Represent attributes (columns).
    - **Diamonds**: Represent relationships between entities.
    - **Lines (Arrows)**: Connect entities and show relationships (one-to-one, one-to-many, many-to-many).

2. write various schema-based constraints

➔ **Schema-Based Constraints**:

- These constraints are directly applied to the schema using Data Definition Language (DDL). They ensure data quality and integrity:

- **Domain Constraints**:
    - Ensure that attribute values belong to a valid domain (e.g., data type restrictions).
- **Key Constraints (Uniqueness Constraints)**:
    - Ensure that each tuple in a relation is unique based on a key attribute (e.g., primary key).
- **Entity Integrity Constraints**:
    - Ensure that primary key attributes cannot have NULL values.
- **Referential Integrity Constraints**:
    - Maintain consistency between related tables (e.g., foreign keys).
- **Check Constraints**:
    - Define custom conditions that must be satisfied (e.g., age 18).