



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No.2

Mapping ER/EER to Relational schema model.

Date of Performance:

Date of Submission:

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:

- Each entity in the ER diagram corresponds to a table in the relational schema.
- The attributes of the entity become the columns of the table.
- The primary key of the entity becomes the primary key of the table.

2. Relationships to Tables:

a. Many-to-Many Relationships:

- Convert each many-to-many relationship into a new table.
- Include foreign key columns in this table to reference the participating entities.
- 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:

- Represented by foreign key columns in one of the participating tables.
- The table on the "many" side of the relationship includes the foreign key column referencing the table on the "one" side.
- The foreign key column typically references the primary key of the related table.

3. Attributes to Columns:

- Each attribute of an entity becomes a column in the corresponding table.
- Choose appropriate data types for each attribute based on its domain and constraints.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

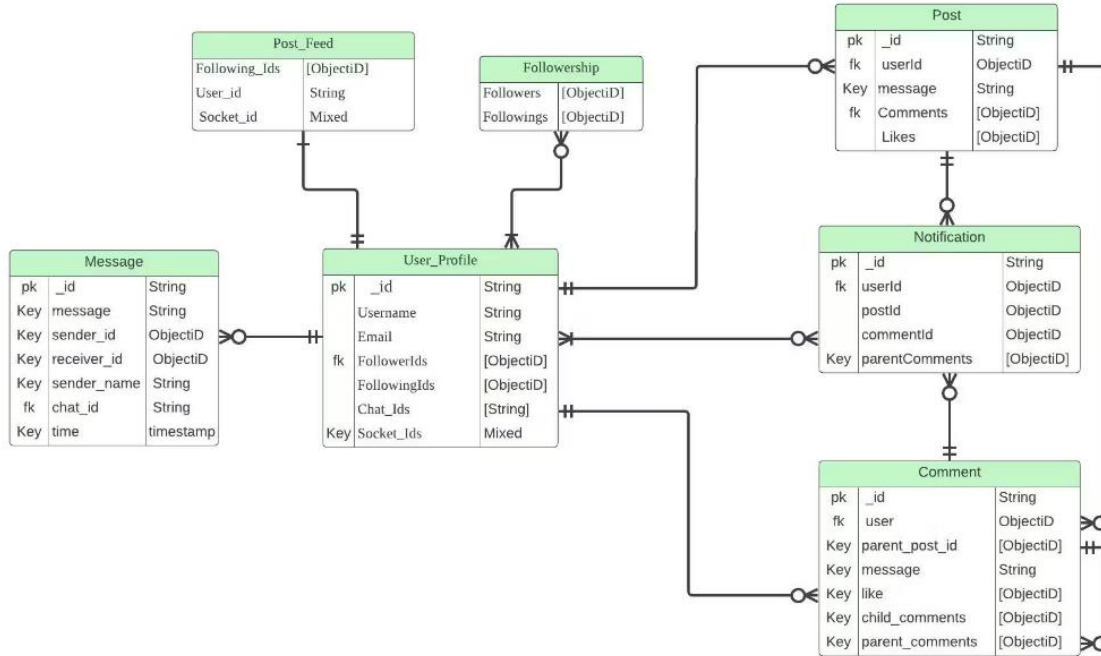
- c. Ensure that attributes participating in relationships are represented as foreign keys when needed.
4. Primary and Foreign Keys:
 - a. Identify the primary key(s) of each table based on the primary key(s) of the corresponding entity.
 - b. Ensure referential integrity by defining foreign keys in 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.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Implementation:



Conclusion:

1. write definition of relational schema and notations.

Schema :

A relational schema is a blueprint or structure that represents the logical arrangement of data elements (tables) and the relationships between them in a relational database. It outlines the tables, their attributes, and the constraints that govern their relationships. The schema provides a formal description of how the database is organized, facilitating data management, querying, and manipulation.

Relational Schema Notations:

1. Tables: Represented by rectangles, each table corresponds to an entity or relation in the database.
 2. Attributes: Shown inside the table rectangles, attributes describe the properties or characteristics of the entities. Each attribute has a name and a data type.
 3. Primary Key: Identified by underlining or highlighting, the primary key uniquely identifies each record within a table. It ensures data integrity and serves as a unique identifier.
 4. Foreign Key: Denoted by an attribute in one table that references the primary key of another table. It establishes relationships between tables, enforcing referential integrity.
 5. Relationships: Illustrated by lines connecting tables, relationships indicate how tables are related to each other. These lines typically represent one-to-one, one-to-many, or many-to-many relationships.
2. write various schema-based constraints
 1. Primary Key Constraint: Enforces unique identification for rows in a table.
 2. Foreign Key Constraint: Maintains relationships between tables to ensure referential integrity.
 3. Unique Constraint: Ensures uniqueness of values in specified columns.
 4. Check Constraint: Imposes conditions on allowable values in a column.
 5. Not Null Constraint: Requires a column to have a value, prohibiting null entries.
 6. Default Constraint: Provides a default value when none is specified.
 7. Referential Integrity Constraint: Ensures validity of relationships between tables.

