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:
    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.

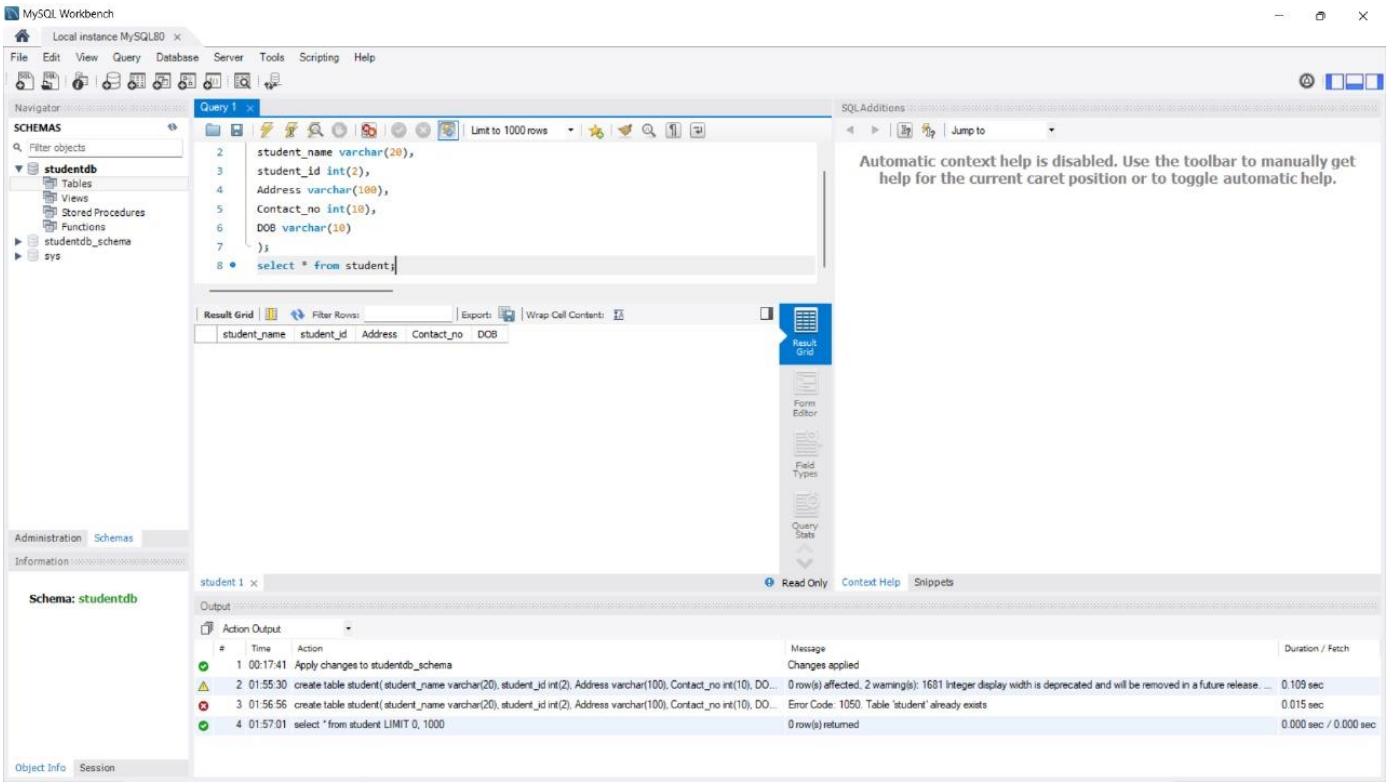
d. Ensure that foreign keys have appropriate constraints, such as ON DELETE CASCADE or ON UPDATE CASCADE, to maintain data integrity.

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:



Conclusion:

1.write definition of relational schema and notations

A relational schema defines the structure of a relational database. It consists of a set of relation schemas, where each relation schema defines the structure of a table (relation) in the database. Each relation schema includes the name of the relation (table) and the names and types of its attributes (columns).

Notations for a relational schema typically include:

Relation (Table) Name: The name of the relation is written at the top, often centered and enclosed in a box or a circle.

Attributes (Columns): Each attribute is listed below the relation name, along with its data type. Attributes are typically written in a column format.

Primary Key: The primary key attribute(s) are underlined to indicate that they uniquely identify each tuple (row) in the relation.

Foreign Key: If an attribute is a foreign key that references another relation, it is typically annotated to indicate the referenced relation and attribute.

Data Types: Data types for each attribute are specified, such as integer, string, date, etc.

Constraints: Any constraints on the attributes, such as NOT NULL or UNIQUE, may be included in the notation.

2.Write various schema-based constraints

Primary Key Constraint: Ensures that a column (or a set of columns) uniquely identifies each record in a table. It enforces the uniqueness and the non-nullability of the specified column(s).

Foreign Key Constraint: Defines a relationship between two tables. It ensures that the values in a column (or a set of columns) of one table match the values in a column (usually the primary key) of another table.
Unique Constraint: Ensures that all values in a column (or a set of columns) are unique across the table.

Unlike the primary key constraint, a unique constraint allows null values, but if a column has a unique constraint, only one null value is allowed.

Check Constraint: Defines a condition that each row must satisfy. It ensures that the values in a column meet specific criteria. For example, a check constraint can be used to ensure that the value in a column is greater than zero.

Not Null Constraint: Ensures that a column cannot have a null value. It requires that every row in the table has a value for that column

.

Default Constraint: Specifies a default value for a column. If no value is provided for the column during an insert operation, the default value is used