UNIT –III

RELATIONAL DATABASE DESIGN AND NORMALIZATION

The process of converting **ER (Entity–Relationship)** and **EER (Enhanced ER)** diagrams into **Relational Schema** is known as **ER/EER-to-Relational Mapping**. This is essential for implementing a conceptual design into an actual database.

The mapping follows a set of systematic rules.

---

1. Mapping ER Model to Relational Schema

Step 1: Mapping Strong Entities

- Each **strong entity** becomes a **table**.
- Attributes become columns.
- **Primary key** of the entity becomes the primary key of the table.

Example

Entity: Student(RollNo, Name, Dept, Age)

Relation:

```
Student(RollNo PRIMARY KEY, Name, Dept, Age)
```

Step 2: Mapping Weak Entities

- Weak entity depends on another entity (owner).
- Its table includes:
  - **Partial key**
  - **Primary key of owner entity** (as foreign key)
- Composite primary key = owner key + partial key.

Example

Weak Entity: Dependent(DepName) dependent on Employee(EmpID)

Relation:

```
Employee(EmpID PRIMARY KEY, Name)
Dependent(EmpID, DepName, PRIMARY KEY(EmpID, DepName),
          FOREIGN KEY(EmpID) REFERENCES Employee)
```

Step 3: Mapping 1–1 Relationship

Two options:

1. Add primary key of one entity as a foreign key in the other.
2. Choose the side with **total participation**.

Example

```
Employee(EmpID, Name, Salary)
Locker(LockerID, Location)
```

1–1 mapping:

```
Employee(EmpID, Name, Salary, LockerID UNIQUE)
```

Step 4: Mapping 1–M (One-to-Many) Relationship

- Add the **primary key of the "one" side** as a **foreign key** on the "many" side.

Example

```
Dept(DeptID, DeptName)
Employee(EmpID, Name, DeptID)
```

Step 5: Mapping M–N (Many-to-Many) Relationship

- Create **new relation** for the relationship.
- Include:
  - Primary keys of both entities
  - Relationship attributes

Example

```
Student(RollNo, Name)
Course(CourseID, Title)
Relationship: Enroll (Grade)
```

Mapping:

```
Enroll(RollNo, CourseID, Grade,
       PRIMARY KEY(RollNo, CourseID),
       FOREIGN KEY(RollNo) REFERENCES Student,
       FOREIGN KEY(CourseID) REFERENCES Course)
```

Step 6: Mapping Multivalued Attributes

- Create a **separate table**.
- Table contains:
  - Primary key of entity
  - Multivalued attribute

Example

Student has multivalued attribute **Phone**.

```
Phone(RollNo, PhoneNo,
      PRIMARY KEY(RollNo, PhoneNo),
```

```
      FOREIGN KEY(RollNo) REFERENCES Student)
```

Step 7: Mapping Composite Attributes

- Use **simple attributes only**.
- Break composite attribute into components.

Example:
Address = (Street, City, Pincode)

Store as:

```
Student(RollNo, Street, City, Pincode)
```

2. Mapping EER Model to Relational Schema

EER extends ER with:

✓ **Specialization / Generalization**
✓ **Inheritance**
✓ **Category (Union Type)**
✓ **Aggregation**

Mapping Specialization / Generalization

Three Common Approaches

**Approach 1: Top-Down / Superclass-Subclass (One Table per Entity)

- Create a table for the **superclass**.
- Create separate tables for each **subclass**, including:
  - Attributes of subclass
  - Primary key of superclass

Example

```
Person(PersonID, Name)
Student(PersonID, Course)
Teacher(PersonID, Salary)
```

**Approach 2: Single Table (One Table for Entire Hierarchy)

- Combine superclass and subclass attributes in **one relation**.
- Use **NULL** where attribute is not applicable.

Example

```
Person(PersonID, Name, Course, Salary)
```

**Approach 3: Only Subclass Tables

- No table for superclass.
- Each subclass table contains:
    - All attributes of superclass
    - All attributes of subclass

Mapping Disjoint vs Overlapping Constraints

- **Disjoint (D)**: entity belongs to only one subclass
- **Overlapping (O)**: entity may belong to multiple subclasses
- **Total / Partial participation** determines if superclass key must appear in subclass.

Mapping Aggregation

Aggregation = relationship between a relationship and an entity.

Mapping:

- Convert the **aggregated relationship** into a separate relation similar to M–N relationship mapping.

Mapping Categories (Union Types)

When a subclass gets its primary key from multiple superclasses.

Mapping:

- Create a table for the **category**.
- Include a **discriminator attribute** to identify which superclass tuple it came from.