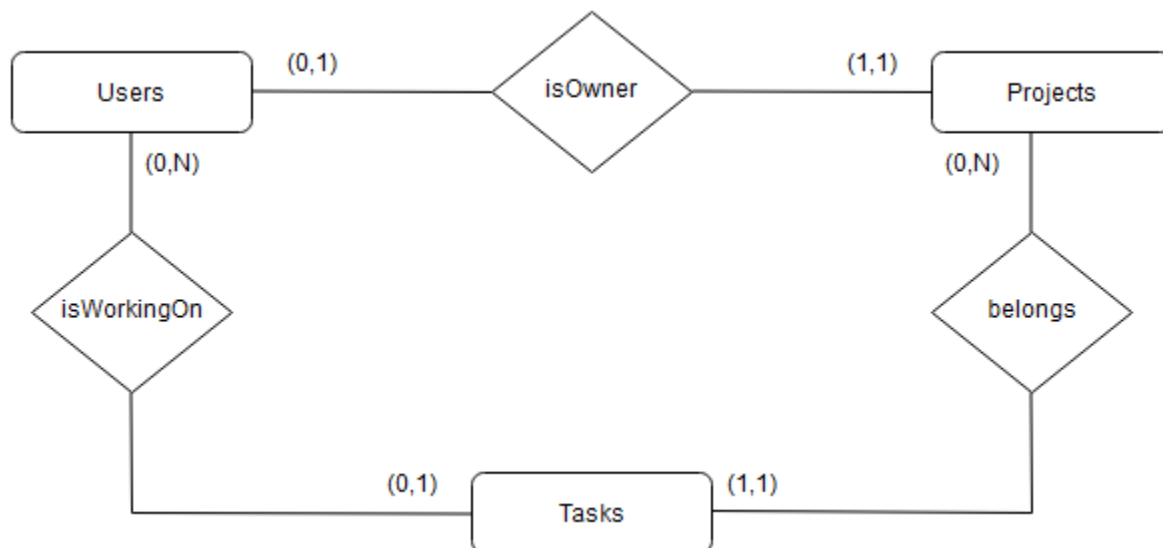


## Building a Task Management System with Entity Framework Code First Approach

### 1. Database Schema:

- Create a SQL Server database named "TaskManagerDB".
- Define the following tables:
  - Users:
    - Id (int, primary key, auto-increment)
    - UserName (varchar(50))
    - Password (varchar(100))
  - Projects:
    - Id (int, primary key, auto-increment)
    - Name (varchar(100))
    - Description (varchar(max))
    - ProjectOwner (foreign key referencing Users.UserId)
  - ToDoTasks:
    - Id (int, primary key, auto-increment)
    - Title (varchar(100))
    - Description (varchar(max))
    - Status (varchar(15))
    - DueDate (datetime)
    - UserId (foreign key referencing Users.UserId)
    - ProjectId (foreign key referencing Projects.ProjectId)

### 2. Relationships:



### **3. Entity Framework Setup:**

- Use Entity Framework Code-First approach to generate the database schema based on the defined entities.
- Set up Entity Framework Core DbContext class named "TaskManagerDbContext" to connect to the "TaskManagerDB" database.
- Define DbSet properties for the User, Project and ToDoTask entities.
- Configure relationships between User, Project and ToDoTask entities using navigation properties and fluent API.

### **4. Entity Classes:**

- Define C# classes representing the User, Project and ToDoTask entities.
- Implement relations between entities using navigation properties.

### **5. Validaton**

- Add validation to ensure that Users.Password contains numbers and special characters
- Ensure that ToDoTasks.Status could contain only Started, In progress and Done values.

### **6. Testing:**

- Test the application thoroughly to ensure all CRUD operations are working correctly.