Time Management system

Part 1: User Requirements Analysis

Description : Time management system is design to non-profit and organizations program. Non-profits often operator with limited resource and rely heavily on volunteers

Task :

* Identify the Target Users: The primary User of the Time Management system are non-profit organizations

1. Administrators : Individuals responsible for overseeing the daily operations of project
2. Project Managers : Leader who plan, execute and monitor team
3. People : who use this system for time management

* Justify Why a Database is Needed for the Application

1. Ensure Data Integrity : maintenance accurate and consistent information
2. Scalability : can handle increasing amounts of data
3. Security : Protect sensitive information by controlling access

Part 2: Translate Requirements to UML

| @startuml ' Define enumeration for Priority enum Priority {  High  Medium  Low }  ' Define enumeration for Status enum Status {  ToDo  InProgress  Completed }  ' User Class class User {  + String ID  + String Name  + String Email  + DateTime CreatedAt  + DateTime UpdatedAt }  ' Task Class class Task {  + String ID  + String Title  + String Description  + Priority Priority  + DateTime Deadline  + Boolean Complete  + Status Status  + DateTime CreatedAt  + DateTime UpdatedAt }  ' Category Class class Category {  + String ID  + String Name  + String Description  + String UserID  + DateTime CreatedAt  + DateTime UpdatedAt }  ' TaskCategory Class (Association Class) class TaskCategory {  + String ID  + String TaskID  + String CategoryID  + DateTime AssignedDate  + Integer Importance }  ' TaskUser Class (Association Class) class TaskUser {  + String ID  + String TaskID  + String UserID  + DateTime AssignedDate }  ' Define Relationships  ' User to Category (One-to-Many) User "1" --> "0..\*" Category : owns  ' Task to TaskCategory (One-to-Many) Task "1" --> "0..\*" TaskCategory : includes  ' Category to TaskCategory (One-to-Many) Category "1" --> "0..\*" TaskCategory : categorized\_by  ' Task to TaskUser (One-to-Many) Task "1" --> "0..\*" TaskUser : assigned\_to  ' User to TaskUser (One-to-Many) User "1" --> "0..\*" TaskUser : assigned\_tasks  @enduml |
| --- |

Explain this diagram

-- Classes from UML mapped to tables, along with One-to-Many relationships:

-- User (1) --> (M) Category

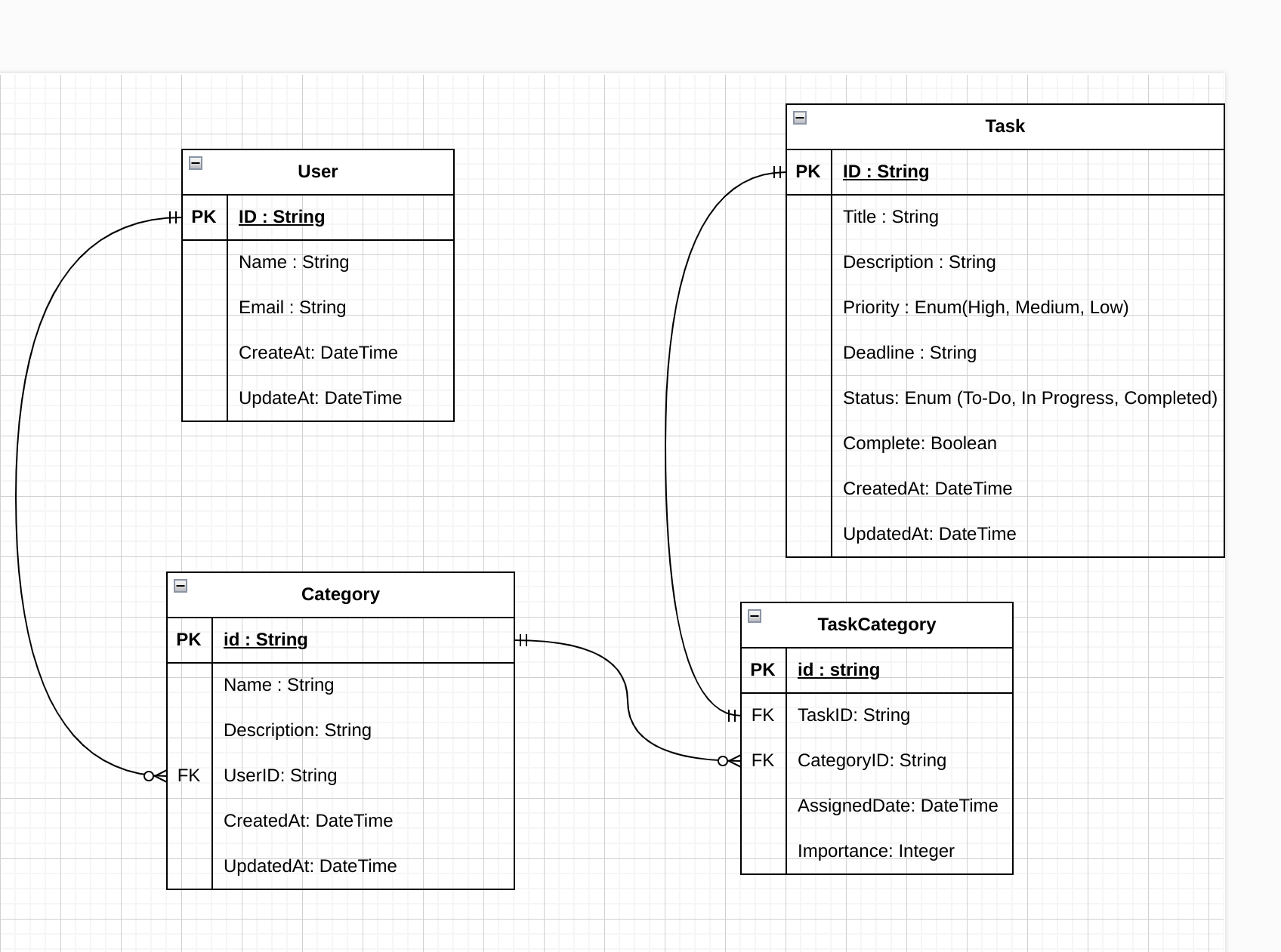
-- Task (1) --> (M) TaskCategory

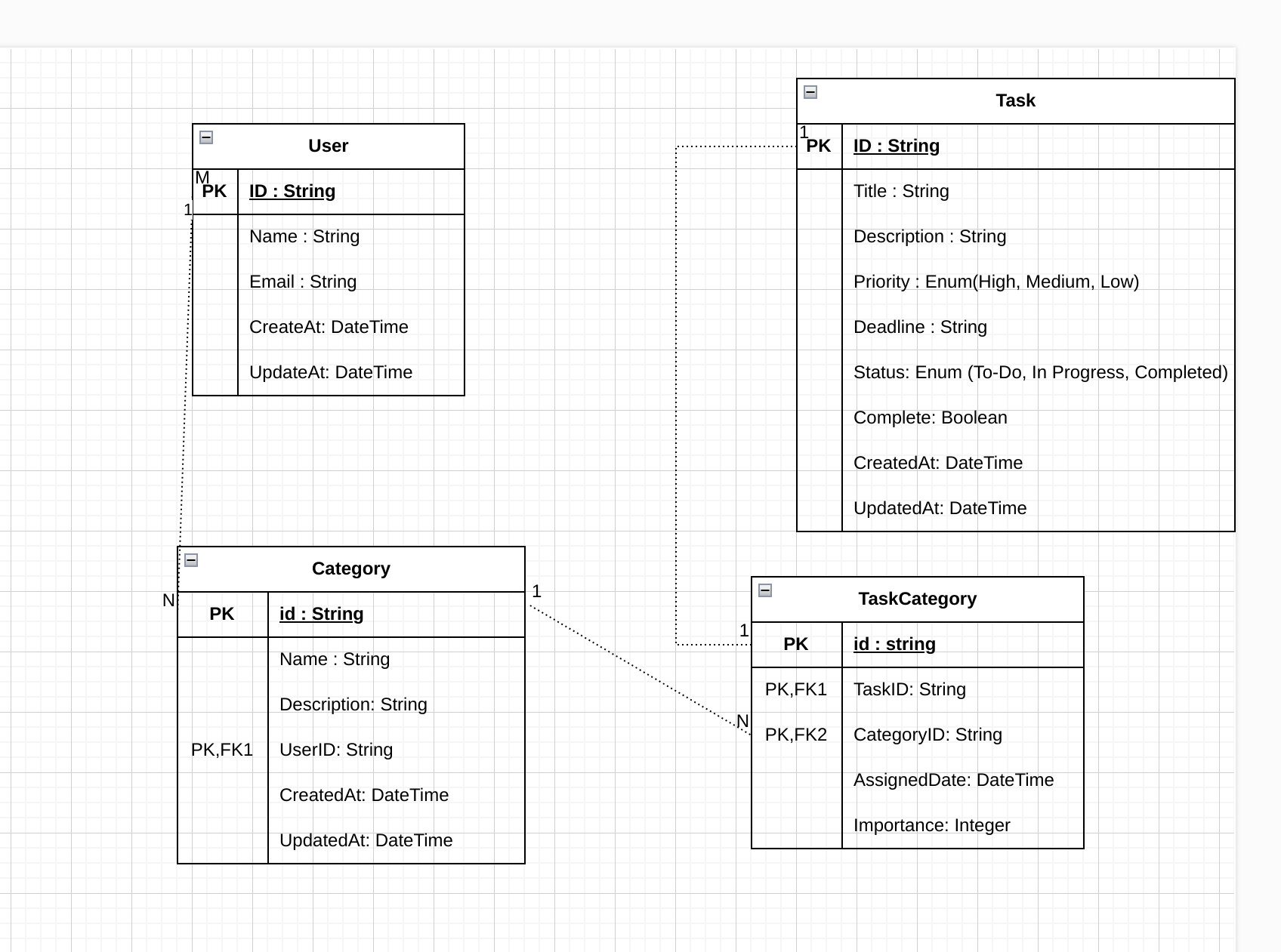
-- Category (1) --> (M) TaskCategory

-- Task (1) --> (M) TaskUser

-- User (1) --> (M) TaskUser

Part 3: UML to Relation





Part 4: Normalization

step by step

1NF from all table have unique id and split data

Apply 1NF to Table

1. User
2. Task
3. Category
4. TaskCategory

analysis:

Atomicity → contain atomic value

Primary key → each table have Primary key

No Repeating Group →There are no repeating groups in any table.

summary this is table pass standard NF1

2 NF from 1NF All and not have Partial dependencies

Apply 2NF to Table

each table

1. User  
   Primary Key : ID  
   Partial dependencies : all non key depend solely on ID (pirmary key)  
   Non-Key Attributes: Title, Description, Priority, Deadline, Complete, Status, CreatedAt, UpdatedAt
2. Task

Primary Key: ID

Non-Key Attributes: Title, Description, Priority, Deadline, Complete, Status, CreatedAt, UpdatedAt

Dependency: All non-key attributes depend solely on ID.

1. Category

Primary Key: ID

Non-Key Attributes: Name, Description, UserID, CreatedAt, UpdatedAt

Dependency: All non-key attributes depend solely on ID.

1. TaskCategory

Primary Key: ID

Non-Key Attributes: TaskID, CategoryID, AssignedDate, Importance

Dependency: All non-key attributes depend solely on ID

3NF From 1NF

write this Check: There are no attributes that depend on other non-key attributes

from each table

Table : user

Primary key : ID

Non-Key Attributes: Name, Email, CreatedAt, UpdatedAt

Dependency: All non-key attributes (Name, Email, CreatedAt, UpdatedAt)

Table : task

Primary Key: ID (ID of task)

Non-Key Attributes: Title, Description, Priority, Deadline, Complete, Status, CreatedAt, UpdatedAt

Dependency: All non-key attributes depend solely on ID.

Table : Category

Primary Key: ID

Non-Key Attributes: Name, Description, UserID, CreatedAt, UpdatedAt

Dependency:

* This table :Name, Description, CreatedAt, and UpdatedAt depend solely on ID.
* Other table :UserID is a foreign key linking to the User table but does not cause any transitive dependency

Table : Taskcategory

Primary Key: ID

Non-Key Attributes: TaskID, CategoryID, AssignedDate, Importance

Dependency: All non-key attributes depend solely on ID.

Table : Taskuser

Primary Key: ID

Non-Key Attributes: TaskID, UserID, AssignedDate

Dependency: All non-key attributes depend solely on ID.

Part 5: Database Implementation using SQL Server

Part 6 : Queries (SQL 5 quereis)

Part 7: Reporting