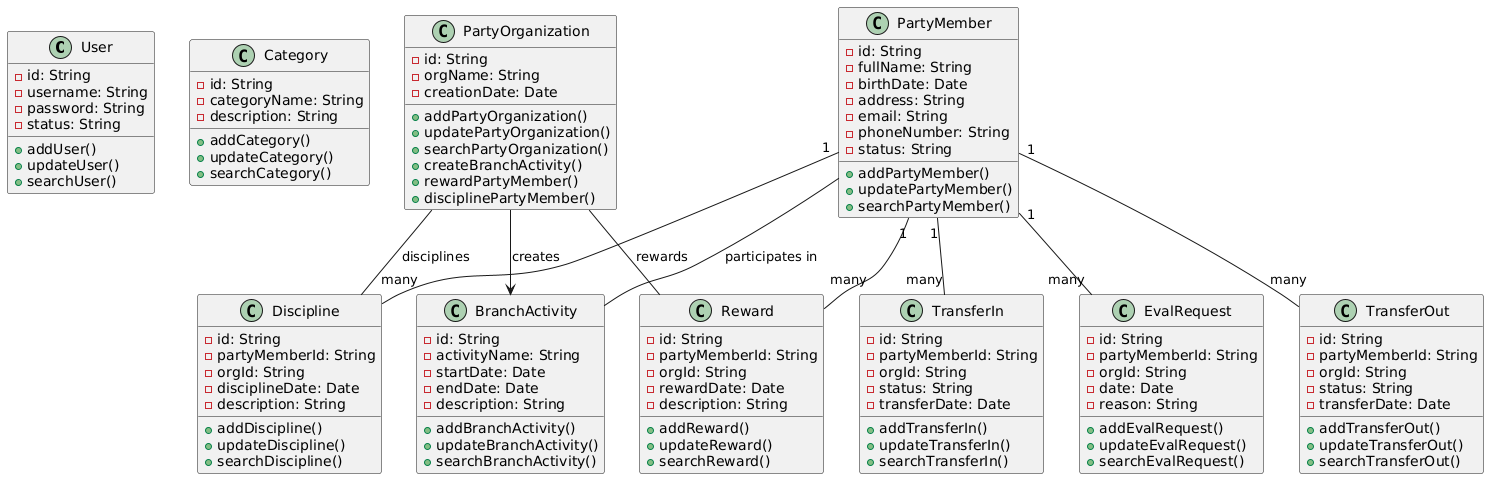
**UML:**



**CSDL SQL**

CREATE DATABASE PartyManagement;

USE PartyManagement;

-- Bảng User

CREATE TABLE User (

id VARCHAR(36) PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(255) NOT NULL,

status VARCHAR(20)

);

-- Bảng Category

CREATE TABLE Category (

id VARCHAR(36) PRIMARY KEY,

categoryName VARCHAR(100) NOT NULL,

description TEXT

);

-- Bảng PartyMember

CREATE TABLE PartyMember (

id VARCHAR(36) PRIMARY KEY,

fullName VARCHAR(100) NOT NULL,

birthDate DATE,

address VARCHAR(255),

email VARCHAR(100),

phoneNumber VARCHAR(20),

status VARCHAR(20)

);

-- Bảng PartyOrganization

CREATE TABLE PartyOrganization (

id VARCHAR(36) PRIMARY KEY,

orgName VARCHAR(100) NOT NULL,

creationDate DATE NOT NULL

);

-- Bảng BranchActivity

CREATE TABLE BranchActivity (

id VARCHAR(36) PRIMARY KEY,

activityName VARCHAR(100) NOT NULL,

startDate DATE NOT NULL,

endDate DATE,

description TEXT,

orgId VARCHAR(36),

FOREIGN KEY (orgId) REFERENCES PartyOrganization(id)

);

-- Bảng TransferOut

CREATE TABLE TransferOut (

id VARCHAR(36) PRIMARY KEY,

partyMemberId VARCHAR(36),

orgId VARCHAR(36),

status VARCHAR(20),

transferDate DATE NOT NULL,

FOREIGN KEY (partyMemberId) REFERENCES PartyMember(id),

FOREIGN KEY (orgId) REFERENCES PartyOrganization(id)

);

-- Bảng TransferIn

CREATE TABLE TransferIn (

id VARCHAR(36) PRIMARY KEY,

partyMemberId VARCHAR(36),

orgId VARCHAR(36),

status VARCHAR(20),

transferDate DATE NOT NULL,

FOREIGN KEY (partyMemberId) REFERENCES PartyMember(id),

FOREIGN KEY (orgId) REFERENCES PartyOrganization(id)

);

-- Bảng EvalRequest

CREATE TABLE EvalRequest (

id VARCHAR(36) PRIMARY KEY,

partyMemberId VARCHAR(36),

orgId VARCHAR(36),

date DATE NOT NULL,

reason TEXT,

FOREIGN KEY (partyMemberId) REFERENCES PartyMember(id),

FOREIGN KEY (orgId) REFERENCES PartyOrganization(id)

);

-- Bảng Reward

CREATE TABLE Reward (

id VARCHAR(36) PRIMARY KEY,

partyMemberId VARCHAR(36),

orgId VARCHAR(36),

rewardDate DATE NOT NULL,

description TEXT,

FOREIGN KEY (partyMemberId) REFERENCES PartyMember(id),

FOREIGN KEY (orgId) REFERENCES PartyOrganization(id)

);

-- Bảng Discipline

CREATE TABLE Discipline (

id VARCHAR(36) PRIMARY KEY,

partyMemberId VARCHAR(36),

orgId VARCHAR(36),

disciplineDate DATE NOT NULL,

description TEXT,

FOREIGN KEY (partyMemberId) REFERENCES PartyMember(id),

FOREIGN KEY (orgId) REFERENCES PartyOrganization(id)

);

-- Bảng trung gian PartyMember\_BranchActivity để quản lý nhiều-nhiều giữa PartyMember và BranchActivity

CREATE TABLE PartyMember\_BranchActivity (

partyMemberId VARCHAR(36),

branchActivityId VARCHAR(36),

PRIMARY KEY (partyMemberId, branchActivityId),

FOREIGN KEY (partyMemberId) REFERENCES PartyMember(id),

FOREIGN KEY (branchActivityId) REFERENCES BranchActivity(id)

);