Солтисюк Дмитро, студент ТУ-12-21-ІПЗ

Модульна контрольна робота

Завдання

Обрано варіант 6:

> Розробити консольний додаток для генерації і початкового наповнення БД системи обмеження доступу.

Отож, завдання полягає у тому, щоб розробити систему обмеження доступу, що включає розробку схеми БД та консольного додатку на C# для генерації і початкового наповнення даними.

Виконання

Отримавши вимоги до створення системи бази даних для обмеження доступу, ми розробили ретельно спроектовану систему, яка має на меті забезпечити безпеку, структурованість та ефективне управління доступом користувачів до різних об’єктів у системі.

Загальний опис системи

Наша система бази даних для обмеження доступу містить набір таблиць, які взаємодіють між собою, щоб забезпечити ефективне управління доступом. Вона включає в себе такі ключові складові:

- **Користувачі (Users)**: Ця таблиця містить інформацію про користувачів системи, які мають доступ до різних об’єктів. Кожен користувач має унікальний ідентифікатор, ім’я та захешований пароль.
- **Ролі (Roles)**: Ролі визначають групи користувачів зі схожими правами доступу. Ця таблиця містить перелік ролей з унікальним ідентифікатором та назвою.
- **Права доступу (Permissions)**: Права доступу визначають конкретні дії, які можуть бути виконані з об’єктами. Вони можуть бути, наприклад, "читання", "запису" чи "видалення". Кожне право доступу має унікальний ідентифікатор та назву.
- **Об’єкти (Objects)**: Об’єкти представляють собою різні елементи системи, до яких можуть бути надані права доступу. Наприклад, це можуть бути файли, директорії, документи тощо. Кожен об’єкт має унікальний ідентифікатор та назву.
- **Зв’язки між сутностями**: Щоб забезпечити функціональність системи, ми створили додаткові таблиці для відображення зв’язків між користувачами та їх ролями, ролями та правами доступу, а також між користувачами, об’єктами та правами доступу.

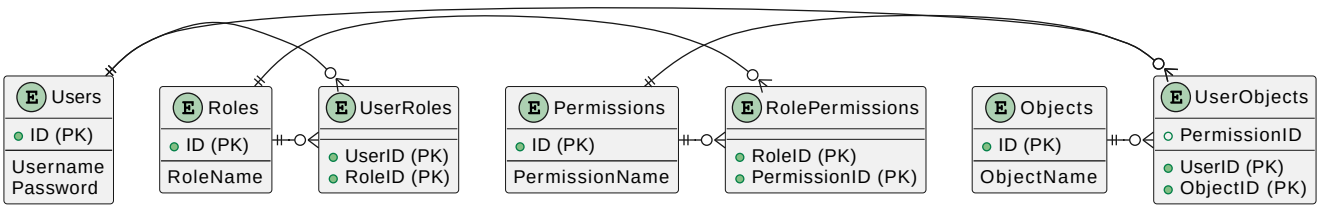
Ця система бази даних ретельно спроектована для забезпечення безпеки, ефективного управління доступом та структурування даних в інформаційній системі.

Для чого це потрібно

1. **Забезпечення безпеки**: Система бази даних для обмеження доступу допомагає захистити конфіденційні дані від несанкціонованого доступу шляхом розподілу прав доступу між користувачами.
2. **Ефективне управління доступом**: За допомогою ролей та прав доступу можна ефективно контролювати, які користувачі мають доступ до конкретних об'єктів та які дії вони можуть виконувати.
3. **Структурування даних**: Ретельно спроектована структура бази даних дозволяє зберігати та організовувати дані таким чином, щоб вони були легко доступні та зрозумілі для подальшого використання.
4. **Скорочення ризику помилок**: Використання бази даних допомагає уникнути помилок, пов'язаних з неправильним управлінням доступом, оскільки права доступу можна легко налаштувати та змінити в одному місці.

Отже, система бази даних для обмеження доступу є ключовою складовою для забезпечення безпеки та ефективного управління доступом у будь-якій інформаційній системі.

Опис системи бази даних



Система бази даних для обмеження доступу призначена для управління правами доступу користувачів до різних об'єктів. У системі існують такі сутності:

- **Користувачі (Users)**: представляють користувачів системи з унікальним ідентифікатором, ім'ям та паролем.
- **Ролі (Roles)**: представляють ролі користувачів, які визначають їхні права доступу. Роль має унікальний ідентифікатор та назву.
- **Права доступу (Permissions)**: визначають можливості доступу користувачів до об'єктів. Право доступу має унікальний ідентифікатор та назву.
- **Об'єкти (Objects)**: це різні об'єкти в системі, до яких можуть бути надані права доступу. Об'єкт має унікальний ідентифікатор та назву.

Схема таблиць

Таблиця `Users`*:*

- `UserId (INT, PRIMARY KEY)` - унікальний ідентифікатор користувача

- `Username (NVARCHAR(50))` - ім’я користувача
- `Password (NVARCHAR(50))` - пароль користувача

Таблиця `Roles`*:*

- `RoleId (INT, PRIMARY KEY)` - унікальний ідентифікатор ролі
- `RoleName (NVARCHAR(50))` - назва ролі

Таблиця `Permissions`*:*

- `PermissionId (INT, PRIMARY KEY)` - унікальний ідентифікатор права доступу
- `PermissionName (NVARCHAR(50))` - назва права доступу

Таблиця `Objects`*:*

- `ObjectId (INT, PRIMARY KEY)` - унікальний ідентифікатор об’єкта
- `ObjectName (NVARCHAR(50))` - назва об’єкта

Таблиця `UserRoles`*:*

- `UserId (INT, FOREIGN KEY)` - ідентифікатор користувача
- `RoleId (INT, FOREIGN KEY)` - ідентифікатор ролі

Таблиця `RolePermissions`*:*

- `RoleId (INT, FOREIGN KEY)` - ідентифікатор ролі
- `PermissionId (INT, FOREIGN KEY)` - ідентифікатор права доступу

Таблиця `UserObjectPermissions`*:*

- `UserId (INT, FOREIGN KEY)` - ідентифікатор користувача
- `ObjectId (INT, FOREIGN KEY)` - ідентифікатор об’єкта
- `PermissionId (INT, FOREIGN KEY)` - ідентифікатор права доступу

Код програми

```
using System;
using Microsoft.Data.Sqlite;

class Program
{
    static void Main(string[] args)
    {
        string connectionString = "Data Source=lab6.db";

        // Підключення до бази даних
        using (SqliteConnection connection = new SqliteConnection(connectionString))
        {
            connection.Open();

            // Створення таблиць та початкове наповнення даними
```

```
            CreateAndPopulateDatabase(connection);
            OutputAllData(connection);

            Console.WriteLine("Базу даних успішно створено та наповнено початковими
даними.");
        }
    }

    static void CreateAndPopulateDatabase(SqliteConnection connection)
    {
        // Створення таблиць із відповідними полями
        string createUserTableQuery = "CREATE TABLE Users (UserId INTEGER PRIMARY KEY,
Username TEXT, Password TEXT)";
        string createRoleTableQuery = "CREATE TABLE Roles (RoleId INTEGER PRIMARY KEY,
RoleName TEXT)";
        string createPermissionTableQuery = "CREATE TABLE Permissions (PermissionId
INTEGER PRIMARY KEY, PermissionName TEXT)";
        string createObjectTableQuery = "CREATE TABLE Objects (ObjectId INTEGER
PRIMARY KEY, ObjectName TEXT)";
        string createUserRoleTableQuery = "CREATE TABLE UserRoles (UserId INTEGER,
RoleId INTEGER, FOREIGN KEY(UserId) REFERENCES Users(UserId), FOREIGN KEY(RoleId)
REFERENCES Roles(RoleId))";
        string createRolePermissionTableQuery = "CREATE TABLE RolePermissions (RoleId
INTEGER, PermissionId INTEGER, FOREIGN KEY(RoleId) REFERENCES Roles(RoleId), FOREIGN
KEY(PermissionId) REFERENCES Permissions(PermissionId))";
        string createUserObjectPermissionTableQuery = "CREATE TABLE
UserObjectPermissions (UserId INTEGER, ObjectId INTEGER, PermissionId INTEGER, FOREIGN
KEY(UserId) REFERENCES Users(UserId), FOREIGN KEY(ObjectId) REFERENCES
Objects(ObjectId), FOREIGN KEY(PermissionId) REFERENCES Permissions(PermissionId))";

        ExecuteQuery(connection, createUserTableQuery);
        ExecuteQuery(connection, createRoleTableQuery);
        ExecuteQuery(connection, createPermissionTableQuery);
        ExecuteQuery(connection, createObjectTableQuery);
        ExecuteQuery(connection, createUserRoleTableQuery);
        ExecuteQuery(connection, createRolePermissionTableQuery);
        ExecuteQuery(connection, createUserObjectPermissionTableQuery);

        // Наповнення таблиць даними
        // Наповнення таблиці Users
        ExecuteQuery(connection, "INSERT INTO Users (UserId, Username, Password)
VALUES (1, 'user1', 'password1')");
        ExecuteQuery(connection, "INSERT INTO Users (UserId, Username, Password)
VALUES (2, 'user2', 'password2')");

        // Наповнення таблиці Roles
        ExecuteQuery(connection, "INSERT INTO Roles (RoleId, RoleName) VALUES (1,
'admin')");
        ExecuteQuery(connection, "INSERT INTO Roles (RoleId, RoleName) VALUES (2,
'user')");
```

```
        // Наповнення таблиці Permissions
        ExecuteQuery(connection, "INSERT INTO Permissions (PermissionId,
PermissionName) VALUES (1, 'read')");
        ExecuteQuery(connection, "INSERT INTO Permissions (PermissionId,
PermissionName) VALUES (2, 'write')");

        // Наповнення таблиці Objects
        ExecuteQuery(connection, "INSERT INTO Objects (ObjectId, ObjectName) VALUES
(1, 'object1')");
        ExecuteQuery(connection, "INSERT INTO Objects (ObjectId, ObjectName) VALUES
(2, 'object2')");

        // Наповнення таблиці UserRoles
        ExecuteQuery(connection, "INSERT INTO UserRoles (UserId, RoleId) VALUES (1,
1)");
        ExecuteQuery(connection, "INSERT INTO UserRoles (UserId, RoleId) VALUES (2,
2)");

        // Наповнення таблиці RolePermissions
        ExecuteQuery(connection, "INSERT INTO RolePermissions (RoleId, PermissionId)
VALUES (1, 1)");
        ExecuteQuery(connection, "INSERT INTO RolePermissions (RoleId, PermissionId)
VALUES (2, 2)");

        // Наповнення таблиці UserObjectPermissions
        ExecuteQuery(connection, "INSERT INTO UserObjectPermissions (UserId, ObjectId,
PermissionId) VALUES (1, 1, 1)");
        ExecuteQuery(connection, "INSERT INTO UserObjectPermissions (UserId, ObjectId,
PermissionId) VALUES (2, 2, 2)");
    }

    static void ExecuteQuery(SqliteConnection connection, string query)
    {
        using (SqliteCommand command = new SqliteCommand(query, connection))
        {
            command.ExecuteNonQuery();
        }
    }

    static void OutputAllData(SqliteConnection connection)
    {
        // Виведення даних з таблиці Users
        OutputTableData(connection, "Users");

        // Виведення даних з таблиці Roles
        OutputTableData(connection, "Roles");

        // Виведення даних з таблиці Permissions
        OutputTableData(connection, "Permissions");

        // Виведення даних з таблиці Objects
```

```
        OutputTableData(connection, "Objects");

        // Виведення даних з таблиці UserRoles
        OutputTableData(connection, "UserRoles");

        // Виведення даних з таблиці RolePermissions
        OutputTableData(connection, "RolePermissions");

        // Виведення даних з таблиці UserObjectPermissions
        OutputTableData(connection, "UserObjectPermissions");
    }

    static void OutputTableData(SqliteConnection connection, string tableName)
    {
        Console.WriteLine($"Дані з таблиці \"{tableName}\":");

        // Retrieve column names
        var columnNames = new List<string>();
        using (var command = new SqliteCommand($"PRAGMA table_info({tableName})",
connection))
        {
            using (var reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    columnNames.Add(reader.GetString(1)); // Column name is at index 1
                }
            }
        }

        // Output column names
        foreach (var columnName in columnNames)
        {
            Console.Write($"{columnName,-20}"); // Adjust spacing as needed
        }
        Console.WriteLine();

        // Retrieve and output data
        using (SqliteCommand command = new SqliteCommand($"SELECT * FROM {tableName}",
connection))
        {
            using (SqliteDataReader reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    for (int i = 0; i < reader.FieldCount; i++)
                    {
                        Console.Write($"{reader[i],-20}"); // Adjust spacing as needed
                    }
                    Console.WriteLine();
                }
```

```
            }
        }
        Console.WriteLine();
    }
}
```

```
MSBuild version 17.9.8+b34f75857 for .NET
  Determining projects to restore...
  All projects are up-to-date for restore.
  6 -> /Users/dmytro.soltusyuk/Work/university/Бази_даних/6/bin/Debug/net8.0/6.dll

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:00.46
Дані з таблиці "Users":
UserId              Username            Password
1                   user1               password1
2                   user2               password2

Дані з таблиці "Roles":
RoleId              RoleName
1                   admin
2                   user

Дані з таблиці "Permissions":
PermissionId        PermissionName
1                   read
2                   write

Дані з таблиці "Objects":
ObjectId            ObjectName
1                   object1
2                   object2

Дані з таблиці "UserRoles":
UserId              RoleId
1                   1
2                   2

Дані з таблиці "RolePermissions":
RoleId              PermissionId
1                   1
2                   2

Дані з таблиці "UserObjectPermissions":
UserId              ObjectId            PermissionId
1                   1                   1
2                   2                   2

Базу даних успішно створено та наповнено початковими даними.
```

Висновок

У результаті виконання модульної контрольної роботи була розроблена гнучка система обмеження доступу з можливістю контролю прав користувача через базу данних. Також

було розроблено консольну програму на мові C#, яка генерує випадкові дані для користувачів, ролей, прав доступу та об’єктів і здійснює наповнення бази даних цими даними для системи обмеження доступу.