

Технічна документація - Bitbucket System (EF Core)

Студент: Луханін Богдан Сергійович , 244(Б)

Дисципліна: Об'єктно-орієнтоване програмування (C#)

Стек: .NET 9.0, C# 13

p.s. : щодо "DataSet-Bitbucket.sql", чесно, не зрозумів чи завдання вимагало створення бази в SSMS, тому обрав підхід code-first, щоб була можливість розгорнути БД на будь-якій машині без втручання в SSMS

1. Вступ та Архітектура

Проект реалізує базу даних для системи контролю версій Bitbucket.

Використан підхід Code-First, де структура бази даних генерується на основі класів

E/R схема та моделі

Реалізовано 5 сутностей з суворою типізацією відповідно до завдання (використання атрибутів `[Column]`):

1. Users

- `Username` , `Password` : `VARCHAR(30)` — для економії пам'яті (замість стандартного `NVARCHAR`).
- `Email` : `VARCHAR(50)`

2. Repositories

- `Name` : `VARCHAR(50)`
- **Зв'язок:** Many-to-Many з Users (реалізовано через `UsingEntity<RepositoriesContributors>`).

3. Issues

- `Title` : `VARCHAR(255)`
- `IssueStatus` : `VARCHAR(6)` (обмеження довжини).

- **Зв'язок:** Зовнішні ключі на Repository та Assignee (User).

4. Commits

- Пов'язує Repository, Contributor (User) та опціонально Issue

5. Files

- **Size** : `DECIMAL(15,2)` - вимога по точності до 2 знаків
- **ParentId** : **Self-Referencing** (файл може бути папкою для іншого файлу).

2. Маніпуляція даними

У класі `Program.cs` реалізовано методи для наповнення та зміни даних

Seeding (Початкове наповнення)

Оскільки SQL-скрипт `DataSet-Bitbucket.sql` не використовувався, реалізовано метод `SeedBaseData()`, який програмно створює:

- 10 користувачів (`User_1` ... `User_10`).
- 10 репозиторіїв та історію коммітів
- Базову файлову структуру для перевірки рекурсивних запитів

Виконання завдань

- **Insert:** Метод `InsertSpecificData()` додає тестові файли (`Trade.idk` , `Controller.php`) та Issues, як вимагається в завданні.
- **Update:** Реалізовано логіку закриття задач (`IssueStatus = 'closed'`) для користувача з `Id = 6`
- **Delete:** Реалізовано видалення репозиторіїв `"Softuni-Teamwork"` разом із його коммітами та задачами для збереження цілісності бази даних

3. Реалізація Запитів

Всі запити реалізовано через LINQ:

Список реалізованих вибірок:

Назва запиту	Умова	Реалізація LINQ
Commits	Всі коміти, сортування за 4 полями	<code>.OrderBy(c => c.Id).ThenBy(...)</code>
Front-end	Файли > 1000 байт та містять "html"	<code>.Where(f => f.Size > 1000 && f.Name.Contains("html"))</code>
Issues	Формат <code>User : Title</code>	<code>.Select(i => new { i.Assignee.Username, i.Title })</code>
Single Files	Файли, які не є папками (немає в ParentId)	<code>.Where(f => !parentIds.Contains(f.Id))</code>
Top Repos	Топ-5 репозиторіїв за активністю	<code>.OrderByDescending(r => r.Commits.Count).Take(5)</code>
Avg Size	Середній розмір файлів юзера	<code>.Average(f => f.Size)</code>

4. Програмованість

- Function `udf_AllUserCommits:C#`

Реалізовував як C# метод, який приймає username і повертає int (== кількість коммітів).

```
context.Commits.Count(c => c.Contributor.Username == userName);
```

- Procedure `usp_SearchForFiles`:

Реалізовано як метод пошуку файлів за розширенням (наприклад, "html") з форматуванням виводу (додавання суфікса "кв")

Висновки

Лабораторна робота покриває всі аспекти роботи з реляційними базами даних: від проектування схеми DDL до запитів/програмованості . А використання Entity Framework Core версії 9.0 дозволило реалізувати завдання швидко та безпечно, дотримуючись принципів ООП.

output:

```
=== STARTING BITBUCKET SYSTEM (FULL LAB) ===
Seeding base data (Users, Repos, Commits)...
Base data seeded.
```

=== SECTION 2: DML ===

Inserting data (task 2)

Specific Files and Issues added.

[Update]]: Closed 0 issues for Assignee ID 6

[Delete]: Repository 'Softuni-Teamwork' deleted.

=== SECTION 3: QUERIES ===

--- 1. Commits Sorted ---

1 | Commit 1 | Repo:1

2 | Commit 2 | Repo:2

3 | Commit 3 | Repo:3

4 | Commit 4 | Repo:4

5 | Commit 5 | Repo:5

--- 2. Front-end Files ---

Index.html (1500.50)

About.html (1200.00)

--- 3. Issues Assignment ---

User_8 : Unreachable code in Index.cs

User_2 : Implement documentation for UsersService.cs

User_3 : Typo fix in Judge.html

User_4 : Critical Problem with HomeController.cs file

--- 4. Single Files --

6 | ParentFile_6.txt | 106.00 KB

8 | ParentFile_8.txt | 108.00 KB

9 | ParentFile_9.txt | 109.00 KB

10 | ParentFile_10.txt | 110.00 KB

11 | Index.html | 1500.50 KB

-- 5. Top Repositories ---

Repo_1 - 1 commits

Repo_2 - 1 commits

Repo_3 - 1 commits

Repo_4 - 1 commits

Repo_5 - 1 commits

--- 6. Users Average File Size ---

User_6 - 7070.44

User_5 - 5031.43

User_7 - 3884.96

User_4 - 3728.58

User_2 - 3513.44

=== SECTION 4: PROGRAMMABILITY ===

---- Function: User Commit Count ---

user 'Bohdan' has 0 commits

-- Procedure: Search files by extension --

Searching for 'html':

11 | Index.html | 1500.50 KB

12 | About.html | 1200.00 KB

!!! ALL TASKS COMPLETED !!!

Process finished with exit code 0.