**ВЕЛИКОТЪРНОВСКИ УНИВЕРСИТЕТ**

**„СВ. СВ. КИРИЛ И МЕТОДИЙ“**



Курсова работа

по

Информационни системи на тема:

**Форум за програмисти реализиран чрез рекурсивна релационна връзка**

Изготвил: Димитър Лъчезаров Захариев

Специалност: Софтуерно инженерство

Фак. №: 20023

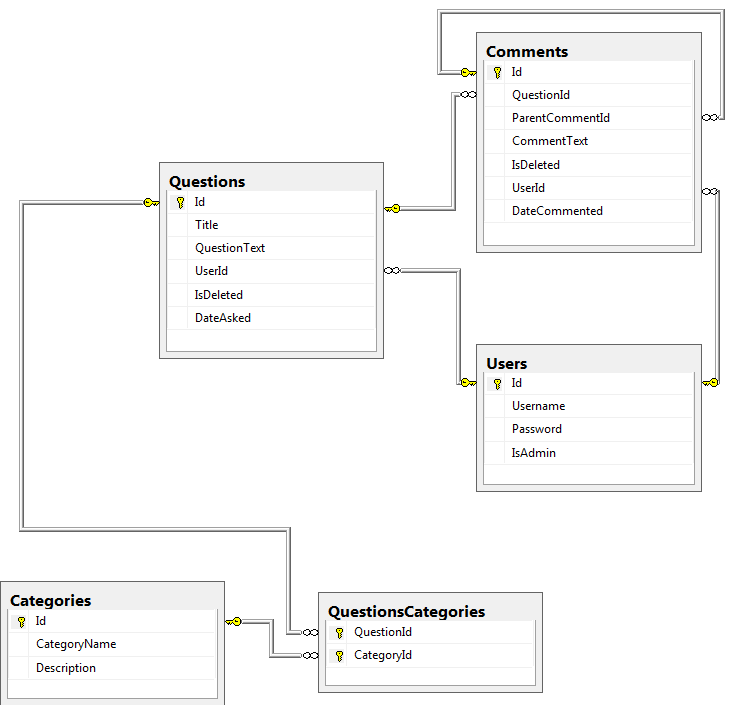
Курс: Втори

Проверил: доц. д-р Цветанка Любомирова Георгиева-Трифонова

1. Предметна област:

Базата от данни описва онлайн форум за програмисти. Следва да бъде използвана от уеб базиран интерфейс.

1. ER-модел на базата от данни



1. Релационен модел на базата от данни

**Users** (Id, Username, Password, IsAdmin)

**Questions** (Id, Title, QuestionText, UserId, IsDeleted, DateAsked)

**Categories** (Id, CategoryName, Description)

**QuestionsCategories** (QuestionId, CategoryId)

**Comments** (Id, QuestionId, ParentCommentId, CommentText, IsDeleted, UserId, DateCommented)

1. Реализация на релационния модел на базата от данни

CREATE DATABASE ProgrammingForumDb

COLLATE Cyrillic\_General\_CI\_AS;

GO

USE ProgrammingForumDb;

CREATE TABLE Users

(

Id INT IDENTITY PRIMARY KEY,

Username NVARCHAR(255) NOT NULL,

[Password] NVARCHAR(255) NOT NULL,

IsAdmin BIT NOT NULL,

CONSTRAINT UC\_Users UNIQUE (Username)

);

CREATE TABLE Categories

(

Id INT IDENTITY PRIMARY KEY,

CategoryName NVARCHAR(255) NOT NULL,

[Description] NVARCHAR(MAX) NOT NULL,

CONSTRAINT UC\_Categories UNIQUE (CategoryName)

);

CREATE TABLE Questions

(

Id INT IDENTITY PRIMARY KEY,

Title NVARCHAR(MAX) NOT NULL,

QuestionText NVARCHAR(MAX) NOT NULL,

UserId INT NOT NULL,

IsDeleted BIT NOT NULL,

DateAsked DATETIME NOT NULL,

CONSTRAINT FK\_Questions\_Users FOREIGN KEY (UserId)

REFERENCES Users(Id)

);

GO

CREATE TABLE Comments

(

Id INT IDENTITY PRIMARY KEY,

QuestionId INT NOT NULL,

ParentCommentId INT,

CommentText NVARCHAR(MAX) NOT NULL,

IsDeleted BIT NOT NULL,

UserId INT NOT NULL,

DateCommented DATETIME NOT NULL,

CONSTRAINT FK\_Comments\_Questions FOREIGN KEY (QuestionId)

REFERENCES Questions(Id),

CONSTRAINT FK\_Comments\_Comments FOREIGN KEY (ParentCommentId)

REFERENCES Comments(Id),

CONSTRAINT FK\_Comments\_Users FOREIGN KEY (UserId)

REFERENCES Users(Id)

);

GO

CREATE TABLE QuestionsCategories

(

QuestionId INT NOT NULL,

CategoryId INT NOT NULL,

CONSTRAINT PK\_QuestionsCategories PRIMARY KEY(QuestionId, CategoryId),

CONSTRAINT FK\_QuestionsCategories\_Questions FOREIGN KEY (QuestionId)

REFERENCES Questions(Id),

CONSTRAINT FK\_QuestionsCategories\_Categories FOREIGN KEY (CategoryId)

REFERENCES Categories(Id)

);

GO

--Добавяне на данни

INSERT INTO Users(Username, [Password], IsAdmin)

VALUES ('admin', 'adminpass', 1);

INSERT INTO Users(Username, [Password],IsAdmin)

VALUES ('pesho','pesho123', 0);

INSERT INTO Users(Username, [Password], IsAdmin)

VALUES ('ivan', 'ivan123', 0);

INSERT INTO Users(Username, [Password], IsAdmin)

VALUES ('ahmet', 'ahmet132', 0);

INSERT INTO Users(Username, [Password], IsAdmin)

VALUES ('hasan', 'hasancho123', 0);

INSERT INTO Users(Username, [Password], IsAdmin)

VALUES ('gosho', 'gosheto1234567', 0);

INSERT INTO Categories(CategoryName, [Description])

VALUES ('Database', 'A database is an organized collection of data. It is the collection of schemas, tables, queries, reports, views, and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information. Use this category if you have questions about designing a database. If it is about a particular database management system, (e.g., MySQL), please use that category instead.');

INSERT INTO Categories(CategoryName, [Description])

VALUES('MySQL', 'MySQL is a free, open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).');

INSERT INTO Categories(CategoryName, [Description])

VALUES('C#', 'C# (pronounced "C sharp") is an object-oriented programming language that is designed for building a variety of applications that run on the .NET Framework. C# is simple, powerful, type-safe, and object-oriented.');

INSERT INTO Categories(CategoryName, [Description])

VALUES('ASP.NET-MVC', 'The ASP.NET MVC Framework is an open source web application framework and tooling that implements a version of the model-view-controller (MVC) pattern tailored towards web applications and built upon an ASP.NET technology foundation.');

INSERT INTO Categories(CategoryName, [Description])

VALUES('OOP', 'Object-oriented programming is a programming paradigm using "objects": data structures consisting of data fields and methods together with their interactions.');

INSERT INTO Categories(CategoryName, [Description])

VALUES('C++', 'C++ is a general-purpose programming language. It was originally designed as an extension to C, and keeps a similar syntax, but is now a completely different language. Use this tag for questions about code (to be) compiled with a C++ compiler.');

INSERT INTO Categories(CategoryName, [Description])

VALUES('JavaScript', 'JavaScript (not to be confused with Java) is a high-level, dynamic, multi-paradigm, weakly-typed language used for both client-side and server-side scripting. Use this category for questions regarding ECMAScript and its various dialects/implementations (excluding ActionScript and Google-Apps-Script). Unless another category for a framework/library is also included, a pure JavaScript answer is expected.');

INSERT INTO Categories(CategoryName, [Description])

VALUES('Java','Java (not to be confused with JavaScript or JScript) is a general-purpose object-oriented programming language designed to be used in conjunction with the Java Virtual Machine (JVM). "Java platform" is the name for a computing system that has installed tools for developing and running Java programs. Use this category for questions referring to the Java programming language or Java platform tools.');

GO

INSERT INTO Questions

VALUES ('how to calculate next 30 days from current date (MySQL)?', 'i want to get next 30 days record from current date. any suggestion or tip will be appreciated.

SELECT end\_date FROM master\_data

WHERE end\_date BETWEEN DATE\_SUB(CURRENT\_DATE(), INTERVAL 30 DAY) AND CURRENT\_DATE();', 1, 0, GETDATE());

GO

INSERT INTO QuestionsCategories

VALUES (1,1);

INSERT INTO QuestionsCategories

VALUES (1,2);

GO

INSERT INTO Comments

VALUES (1, NULL, 'You need DATE\_ADD not DATE\_SUB, because the next 30 days will be between current date and current date + 30:

SELECT end\_date

FROM master\_data

WHERE end\_date BETWEEN CURRENT\_DATE()

AND DATE\_ADD(CURRENT\_DATE(), INTERVAL 30 DAY);', 0, 2, GETDATE());

INSERT INTO Comments

VALUES (1, NULL, 'You have already done DATE\_SUB, just do DATE\_ADD instead.', 0, 3, GETDATE());

GO

INSERT INTO Comments

VALUES (1, 2, 'no he meant ...', 0, 4, GETDATE());

GO

INSERT INTO Comments

VALUES (1, 3, 'yea but if he does this...', 0, 5, GETDATE());

INSERT INTO Comments

VALUES (1, 1, 'That way he will not be able to...', 0, 5, GETDATE());

INSERT INTO Questions

VALUES ('What is the difference between String and string in C#?', 'Example (note the case):

string s = "Hello world!";

String S = "Hello world!";

What are the guidelines for the use of each? And what are the differences?', 3, 0 , GETDATE());

GO

INSERT INTO QuestionsCategories

VALUES (2, 3);

GO

INSERT INTO Comments

VALUES (2, NULL, 'string is an alias in C# for System.String. So technically, there is no difference. It is like int vs. System.Int32.

As far as guidelines, I think it is generally recommended to use string any time you are referring to an object.

e.g.

string place = "world";

Likewise, I think it is generally recommended to use String if you need to refer specifically to the class.

e.g.

string greet = String.Format("Hello {0}!", place);

This is the style that Microsoft tends to use in their examples.

It appears that the guidance in this area may have changed, as StyleCop now enforces the use of the C# specific aliases', 0, 6, GETDATE());

GO

INSERT INTO Comments

VALUES (2, 6, 'If you decide to use StyleCop and follow that, that will say to use the types specific to the language. So for C# you will have string (instead of String), int (instead of Int32), float (instead of Single)...', 0, 1, GETDATE());

1. Създаване на необходимите тригери и процедури

* Тригер който маркира коментарите към даден въпрос като изтрити, след маркиране на въпроса като изтрит

CREATE TRIGGER Post\_delete ON Questions AFTER UPDATE

AS

BEGIN

UPDATE Comments

SET IsDeleted = 1

WHERE QuestionId =(

SELECT i.Id

FROM inserted i

WHERE i.isDeleted = 1

)

END

* Процедура която маркира въпрос като изтрит

CREATE PROCEDURE udp\_MarkQuestionAsDeleted

@QuestionId INT

AS

BEGIN

UPDATE Questions

SET IsDeleted=1

WHERE Id=@QuestionId

END

EXEC udp\_MarkQuestionAsDeleted @QuestionId='1' изпълняване на процедурата в този случай с QuestionId 1

1. Създаване на заявки

* Да се изкарат коментарите заедно с всички подкоментари към него които не са изтрити (йерархията), на даден въпрос, с отстъп - (в този случай въпрос с Id 1)

WITH UpperHierarchy (CommentId, QuestionId, CommentText, ParentCommentId, DateCommented, UserId, HierarchyOrder,

lineage)

AS (SELECT com.Id,

com.QuestionId,

com.CommentText,

com.ParentCommentId,

com.DateCommented,

com.UserId,

0 AS HierarchyOrder,

Cast ('/' AS VARCHAR(255)) AS Lineage

FROM Comments AS com

WHERE com.ParentCommentId IS NULL AND IsDeleted=0

UNION ALL

(SELECT com.Id,

com.QuestionId,

com.CommentText,

com.ParentCommentId,

com.DateCommented,

com.UserId,

HierarchyOrder + 1,

Cast(lineage + Ltrim(Str(com.ParentCommentId, 6, 0))

+ '/' AS VARCHAR(255))

FROM Comments AS com

INNER JOIN UpperHierarchy AS parent

ON com.ParentCommentId = parent.CommentId

WHERE com.IsDeleted=0))

SELECT CommentTextFormatted, com.CommentId, lineage

FROM Questions AS Q

INNER JOIN

(SELECT Space(HierarchyOrder\*5) + CommentText AS CommentTextFormatted, CommentId, QuestionId, ParentCommentId, DateCommented, UserId, lineage

FROM UpperHierarchy) AS Com

ON Com.QuestionId=Q.Id

INNER JOIN Users AS U

ON U.Id=Com.UserId

WHERE Q.Id= 1 --with id 1

ORDER BY lineage + Ltrim(Str(CommentId, 6, 0))

* Да се маркира коментар и всички негови подкоментари като изтрити (в този случай коментар с Id 1)

WITH UpperHierarchy (Id, QuestionId, CommentText, ParentCommentId, DateCommented, UserId, IsDeleted, HierarchyOrder,

lineage)

AS (SELECT com.Id,

com.QuestionId,

com.CommentText,

com.ParentCommentId,

com.DateCommented,

com.UserId,

com.IsDeleted,

0 AS HierarchyOrder,

Cast ('/' AS VARCHAR(255)) AS Lineage

FROM Comments AS com

WHERE com.Id=1

UNION ALL

(SELECT com.Id,

com.QuestionId,

com.CommentText,

com.ParentCommentId,

com.DateCommented,

com.UserId,

com.IsDeleted,

HierarchyOrder + 1,

Cast(lineage + Ltrim(Str(com.ParentCommentId, 6, 0))

+ '/' AS VARCHAR(255))

FROM Comments AS com

INNER JOIN UpperHierarchy AS parent

ON com.ParentCommentId = parent.Id))

UPDATE Comments

SET IsDeleted=1

WHERE Id IN (SELECT Id FROM UpperHierarchy)

* Да се изкарат коментарите заедно с всички подкоментари към него които не са изтрити (йерархията), на даден въпрос, с отстъп, и потребителят на който принадлежат - (в този случай въпрос с Id 1)

WITH UpperHierarchy (CommentId, QuestionId, CommentText, ParentCommentId, DateCommented, UserId, HierarchyOrder,

lineage)

AS (SELECT com.Id,

com.QuestionId,

com.CommentText,

com.ParentCommentId,

com.DateCommented,

com.UserId,

0 AS HierarchyOrder,

Cast ('/' AS VARCHAR(255)) AS Lineage

FROM Comments AS com

WHERE com.ParentCommentId IS NULL AND IsDeleted=0

UNION ALL

(SELECT com.Id,

com.QuestionId,

com.CommentText,

com.ParentCommentId,

com.DateCommented,

com.UserId,

HierarchyOrder + 1,

Cast(lineage + Ltrim(Str(com.ParentCommentId, 6, 0))

+ '/' AS VARCHAR(255))

FROM Comments AS com

INNER JOIN UpperHierarchy AS parent

ON com.ParentCommentId = parent.CommentId

WHERE com.IsDeleted=0))

SELECT CommentTextFormatted, U.Username, lineage, com.CommentId

FROM Questions AS Q

INNER JOIN

(SELECT Space(HierarchyOrder\*5) + CommentText AS CommentTextFormatted, CommentId, QuestionId, ParentCommentId, DateCommented, UserId, lineage

FROM UpperHierarchy) AS Com

ON Com.QuestionId=Q.Id

INNER JOIN Users AS U

ON U.Id=Com.UserId

WHERE Q.Id=1 --with id 1

ORDER BY lineage + Ltrim(Str(CommentId, 6, 0))

1. Потребителски интерфейс

Интерфейсът съдържа функционалност за регистрация, логин, добавяне на въпрос, изтриване на въпрос, редактиране на въпрос, търсене на въпрос, добавяне на коментар към въпрос, добавяне на коментар към коментар, показване на всички въпроси, показване на въпрос с всички коментари към него, изтриване на коментар, добавяне на категория.

Линк към хранилището на системата - <https://github.com/MitkoZ/ProgrammingForum>

1. Бъдеща работа

Бъдещото развитие на приложението би било добавяне на повече функционалност на потребителите – админи, гласуване за въпроси и отговори (точкова система), приет отговор от питащия въпроса. (този отговор който най-много му е помогнал)