МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине «Разработка приложений баз данных для информационных систем»

# на тему: «Разработка серверной части информационной системы в СУБД *MS SQL Server*»

Выполнил: студент гр. ИТИ-31

*Степченко М.Р.*

Принял: преподаватель

Малиновский И. Л.

Гомель 2023

**Цель работы:** разработать серверную часть клиент-серверной информационной системы, основанной на базе данных в заданной предметной области средствами СУБД *MS SQL Server*.

**Практическое задание**

**Задание.**

1. Разработать логическую модель реляционной базы данных, моделирующую предметную область согласно своему варианту задания. Структура БД должна быть нормализована – таблицы должны удовлетворять требованиям третьей нормальной формы.

2. Создать базу данных и таблицы в СУБД *MS* *SQL* *Server* и заполнить их тестовым набором данных, для этого написать *Transact* *SQL* скрипт:

2.1. Создания базы данных и ее таблиц.

При создании таблиц должны быть назначены первичные и внешние ключи и установлены необходимые ограничения целостности данных. Наименования таблиц и полей давать в соответствии с соглашением об именовании этих объектов.

2.2. Заполнения не менее чем трех таблиц БД данными (см. пример внутри этого электронного курса).

При выборе таблиц для заполнения тестовыми наборами руководствоваться следующим:

выбранные таблицы должны содержать основную информацию, касающиеся предметной области приложения;

таблицы должны быть связаны непосредственно;

следует воздерживаться от выбора таблиц, характеризующих кадровую подсистему приложения (данные о сотрудниках и их должностях);

не менее, чем одна таблица должна находиться в схеме базы данных на стороне отношения «многие».

При этом заполнение осуществлять в следующем порядке: сначала генерируются данные для таблиц на стороне отношения «один» (таблицы- «справочники»), потом – в таблицы на стороне отношения «многие» («оперативные таблицы»).

БД заполнить записями в количестве, необходимом для отладки и демонстрации возможностей приложения. Таблицы на стороне отношения один должны содержать не менее 500 записей, таблице на стороне отношения многие должны содержать не менее 20000 записей.

3. Создать с использованием средств *Transact* *SQL* представления, позволяющие отображать данные в удобном для пользователя виде, и заменяющие часто используемые запросы на выборку из двух и более связанных таблиц.

4. Написать не менее трех хранимых процедур с параметрами для вставки и (или) обновление данных в таблицы базы данных.

Вариант № 25: «Брачное агентство»

Предметная область

1. Знаки зодиака (наименование, описание).

2. Национальности (наименование, замечания).

3. Дополнительные услуги (наименование, описание, цена).

4. Клиенты (фото, ФИО, пол, профессия, дата рождения, возраст, рост, вес, количество детей, семейное положение, вредные привычки, хобби, знак зодиака, национальность, адрес, телефон, паспортные данные, информация об искомом партнёре).

5. Услуги (клиент, дата, перечень оказанных услуг – до трех, стоимость, сотрудник).

**Ход работы**

**Порядок выполнения задания.**

В процессе разработки схемы базы данных было принято решение о создании таблиц *ZodiacSigns*, *Nationalities*, *Clients*, *AllServices*, *Employees*, *ProvidedServices*. На рисунке 1 показана схема созданной базы данных «Брачное Агенство».

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Автоматически созданное описание

Рисунок 1 – Схема базы данных

Для создания базы данных и таблиц был разработан скрипт *CreateDataBase*.*sql,* листинг которого приведён в приложении А. Для заполнения тестовыми данными был разработан скрипт *FillDataBase.sql*, который генерирует 500 записей в таблицу *Clients,* 20000 записей – в таблицу *ProvidedServices.*

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 2 – Данные в таблице *ProvidedServices*

Для выборки данных с таблицы *Employee* и *ProvidedServices* было разработано представление *staff.* На рисунке 3 показаны первые 10 записей вызова данного представления.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 3 – Данные представления *staff*

Для вывода данных о клиентах разработано представление *viewClient*. На рисунке 4 показаны первые 10 записей вызова данного представления.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 4 – Данные представления *viewClient*

Для вывода оказанных услуг было разработано представление *viewServices*. На рисунке 5 показаны первые 10 записей вызова данного представления.



Рисунок 5 – Данные представления *viewServices*

Для обновления поля ServiceID в таблице *ProvidedServices* была создана хранимая процедура *ServiceChange.*

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рисунок 6 – Таблица *ProvidedServices* после выполнения *ServiceChange*

Для добавления нового сотрудника в таблицу Employee была разработана хранимая процедура *AddNewEmployee,* которая добавляет новую запись с именем тематики, переданной в параметре. На рисунке 7 показано выполнения скрипта *AddNewEmployee.*



Рисунок 7 – Проверка корректности выполнения процедур *AddNewEmployee*

Для отображения трёх последних заказов у клинета была разработана хранимая функция SelectLatestServicesByCLient.

Изображение выглядит как текст, Шрифт, линия, число

Автоматически созданное описание

Рисунок 8 – Пример выполнения процедуры SelectLatestServicesByCLient

**Вывод:** в процессе работы ознакомились с разработкой серверной части клиент-серверной информационной системы с использованием *MS* *SQL.*

**Приложение А**

**(обязательное)**

**Листинг программы**

**CreateDataBase.sql:**

CREATE DATABASE TestingSystemDb;

USE TestingSystemDb;

CREATE TABLE Users (

UserID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Login VARCHAR(20) NOT NULL UNIQUE,

Password VARCHAR(30) NOT NULL,

Email VARCHAR(40) NOT NULL UNIQUE,

Name VARCHAR(15) NOT NULL,

Surname VARCHAR(20) NOT NULL

);

CREATE TABLE Difficulties(

DifficultyID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name VARCHAR(20) NOT NULL UNIQUE

);

CREATE TABLE Topics(

TopicID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name VARCHAR(20) NOT NULL UNIQUE

);

CREATE TABLE Tests (

TestID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name VARCHAR(40) NOT NULL,

TopicID INT NOT NULL,

DifficultyID INT NOT NULL,

NumberOfQuestions INT NOT NULL DEFAULT 0,

CompletionMark BIT NOT NULL DEFAULT 0,

FOREIGN KEY(TopicID) REFERENCES Topics(TopicID) ON DELETE CASCADE,

FOREIGN KEY(DifficultyID) REFERENCES Difficulties(DifficultyID) ON DELETE CASCADE

);

CREATE TABLE Results(

ResultID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Mark INT NOT NULL,

TestID INT NOT NULL,

UserID INT NOT NULL,

FOREIGN KEY(TestID) REFERENCES Tests(TestID) ON DELETE CASCADE,

FOREIGN KEY(UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

CREATE TABLE Questions(

QuestionID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Name VARCHAR(20) NOT NULL,

Text VARCHAR(200) NOT NULL,

TestID INT NOT NULL,

NumberOfPoints INT NOT NULL,

FOREIGN KEY(TestID) REFERENCES Tests(TestID) ON DELETE CASCADE

);

CREATE TABLE Answers(

AnswerID INT IDENTITY(1,1) PRIMARY KEY NOT NULL,

Text VARCHAR(100) NOT NULL,

QuestionID INT NOT NULL,

FOREIGN KEY(QuestionID) REFERENCES Questions(QuestionID) ON DELETE CASCADE

);

**FillDataBase.sql:**

Use TestingSystemDb;

INSERT INTO Difficulties(Name)

VALUES

('Лёгкая'),

('Средняя'),

('Тяжёлая')

INSERT INTO Topics(Name)

VALUES

('Игры'),

('Фильмы'),

('Программирование'),

('География'),

('Спорт')

DECLARE @counter INT = 1

DECLARE @numberOfUsers INT = 500

DECLARE @Symbol CHAR(52) = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'

DECLARE @MinNumberSymbol INT

DECLARE @MaxNumberSymbol INT

DECLARE @i INT

DECLARE @Position INT

DECLARE @NameLimit INT

-- Таблица Users

SET @MinNumberSymbol = 5

SET @MaxNumberSymbol = 15

DECLARE @Login VARCHAR(20), @Password VARCHAR(30), @Email VARCHAR(40), @Name VARCHAR(15), @Surname VARCHAR(20)

WHILE @counter <= @numberOfUsers

BEGIN

SET @NameLimit=@MinNumberSymbol+RAND()\*(@MaxNumberSymbol-@MinNumberSymbol)

SET @Login = ''

SET @Password = ''

SET @Email = '' --+ '@gmail.com'

SET @Name = ''

SET @Surname = ''

Set @i=1

WHILE @i<=@NameLimit

BEGIN

SET @Position=RAND()\*52

SET @Login = @Login + SUBSTRING(@Symbol, @Position, 1)

SET @Position=RAND()\*52

SET @Password=@Password + SUBSTRING(@Symbol, @Position, 1)

SET @Position=RAND()\*52

SET @Email=@Email + SUBSTRING(@Symbol, @Position, 1)

SET @Position=RAND()\*52

SET @Name=@Name + SUBSTRING(@Symbol, @Position, 1)

SET @Position=RAND()\*52

SET @Surname=@Surname + SUBSTRING(@Symbol, @Position, 1)

SET @i=@i+1

END

SET @Email = @Email + '@gmail.com'

INSERT INTO Users(Login, Password, Email, Name, Surname)

VALUES (@Login, @Password, @Email, @Name, @Surname)

SET @counter = @counter + 1

END

-- Таблица Tests

SET @counter = 1

SET @MaxNumberSymbol = 40

DECLARE @numberOfTests INT = 4000

DECLARE @TopicID INT, @DifficultyID INT

WHILE @counter <= @numberOfTests

BEGIN

SET @NameLimit=@MinNumberSymbol+RAND()\*(@MaxNumberSymbol-@MinNumberSymbol)

SET @Name = ''

Set @i=1

WHILE @i<=@NameLimit

BEGIN

SET @Position=RAND()\*52

SET @Name = @Name + SUBSTRING(@Symbol, @Position, 1)

SET @i=@i+1

END

SET @TopicID = RAND()\*5+1

SET @DifficultyID = RAND()\*3+1

INSERT INTO Tests(Name, TopicID, DifficultyID)

VALUES (@Name, @TopicID, @DifficultyID)

SET @counter = @counter + 1

END

-- Таблица Questions

SET @MaxNumberSymbol = 20

SET @counter = 1

DECLARE @numberOfQuestions INT = 20000

DECLARE @TestID INT

DECLARE @Text VARCHAR(200)

DECLARE @NumberOfPoints INT

WHILE @counter <= @numberOfQuestions

BEGIN

SET @NameLimit=@MinNumberSymbol+RAND()\*(@MaxNumberSymbol-@MinNumberSymbol)

SET @Name = ''

SET @Text = ''

Set @i=1

WHILE @i<=@NameLimit

BEGIN

SET @Position=RAND()\*52

SET @Name = @Name + SUBSTRING(@Symbol, @Position, 1)

SET @Position=RAND()\*52

SET @Text = @Text + SUBSTRING(@Symbol, @Position, 1)

SET @i=@i+1

END

SET @Text = @Text + '?'

SET @TestID = RAND()\*@numberOfTests+1

SET @NumberOfPoints = RAND()\*5+1

INSERT INTO Questions(Name, Text, TestID, NumberOfPoints)

VALUES (@Name, @Text, @TestID, @NumberOfPoints)

SET @counter = @counter + 1

END

-- Таблица Answers

SET @MaxNumberSymbol = 100

SET @counter = 1

DECLARE @QuestionID INT

DECLARE @AnswerText VARCHAR(100)

WHILE @counter <= @numberOfQuestions

BEGIN

SET @NameLimit=@MinNumberSymbol+RAND()\*(@MaxNumberSymbol-@MinNumberSymbol)

SET @Text = ''

Set @i=1

WHILE @i<=@NameLimit

BEGIN

SET @Position=RAND()\*52

SET @Text = @Text + SUBSTRING(@Symbol, @Position, 1)

SET @i=@i+1

END

SET @QuestionID = @counter

INSERT INTO Answers(Text, QuestionID)

VALUES (@Text, @QuestionID)

SET @counter = @counter + 1

END

-- Таблица Results

DECLARE @UserID INT

DECLARE @Mark INT

DECLARE @numberOfResults INT = 20000

SET @counter = 1

WHILE @counter <= @numberOfResults

BEGIN

SET @Mark = RAND()\*10+1

SET @TestID = RAND()\*@numberOfTests+1

SET @UserID = RAND()\*@numberOfUsers+1

INSERT INTO Results(Mark, TestID, UserID)

VALUES (@Mark, @TestID, @UserID)

SET @counter = @counter + 1

END

**TestsWithNumberOfQuestionsView.sql:**

CREATE VIEW TestsWithNumberOfQuestions AS

SELECT Tests.Name As TestName, Tests.NumberOfQuestions,

Difficulties.Name AS Difficulty,

Topics.Name AS Topic

FROM Tests INNER JOIN Difficulties ON Tests.DifficultyID = Difficulties.DifficultyID

INNER JOIN Topics ON Tests.TopicID = Topics.TopicID

**AnswersWithQuestionsView.sql:**

CREATE VIEW AnswersWithQuestions AS

SELECT Questions.Name As QuestionName, Answers.Text as Answer, Tests.Name as TestName

FROM Questions INNER JOIN Answers ON Questions.QuestionID=Answers.AnswerID

INNER JOIN Tests ON Questions.TestID = Tests.TestID

**NumberOfResultsView.sql:**

CREATE VIEW NumberOfResults AS

SELECT Users.Name, Users.Surname,

(SELECT COUNT(\*) FROM Results WHERE Results.UserID=Users.UserID) AS NumberOfResults

FROM Users

**UpdateNumberOfQuestions.sql:**

CREATE PROCEDURE UpdateNumberOfQuestions

AS UPDATE Tests

SET NumberOfQuestions = (SELECT COUNT(\*) FROM Questions WHERE Questions.TestID=Tests.TestID)

**CheckFirstProc.sql:**

EXEC UpdateNumberOfQuestions

**AddTopic.sql:**

CREATE PROCEDURE AddTopic

@name VARCHAR(20)

AS

INSERT INTO Topics(Name)

VALUES(@name)

**CheckSecondProc.sql:**

EXEC AddTopic 'TestTopic'

SELECT \* FROM Topics WHERE NAME=’TestTopic’

**UpdateAnswer.sql:**

CREATE PROCEDURE UpdateAnswer

@AnswerID INT,

@NewAnswer VARCHAR(100)

AS UPDATE Answers

SET Text=@NewAnswer

WHERE Answers.AnswerID=@AnswerID

**CheckThirdProc.sql:**

EXEC UpdateAnswer 9, 'TestAnswer'

SELECT \* FROM Answers WHERE AnswerID=9