Практика по базам данных

ОТЧЕТ

Калюжный Константин

241 группа

Предметная область: «Страноведение»

Содержание

ОПИСАНИЕ СИСТЕМЫ 2

Требования 2

Модель данных 2

Функциональность 3

Серверная часть 3

Клиентская часть 4

СКРИПТЫ 5

Серверная часть 5

Хранимые процедуры и функции 5

Триггеры 6

Представления 6

Клиентская часть 7

Создание и заполнение базы данных 9

**ОПИСАНИЕ СИСТЕМЫ**

**Требования**

Система, о которой идет речь, предназначена для туристического клуба. Она должна хранить информацию о странах мира – для помощи в составлении маршрута, культурной программы поездки и т.д.

Администратор добавляет в базу данные о географических и культурных объектах, чтобы пользователь смог получить ответы на подобного рода вопросы:

* “Где расположены самые длинные реки?”
* “В каких театрах стоит побывать, если понимаешь французский язык?”

Для предоставления качественной и полной информации база должна содержать в себе следующие данные:

* Страны (государственные языки, площадь, количество населения)
* Интересные природные объекты (реки, озера, горы)
* Города ( музеи, театры, значительные архитектурные объекты и другого рода достопримечательности.

Если хранящийся в базе объект включен в список объектов всемирного наследия ЮНЕСКО, это обязательно должно быть отмечено.

**Модель данных**



**Функциональность**

**Серверная часть**

|  |  |  |
| --- | --- | --- |
| ***Хранимые процедуры/функции*** | *Реализация* | *Комментарии* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| ***Триггеры*** | *Реализация* | *Комментарии* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| ***Представления*** | *Реализация* | *Комментарии* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Клиентская часть**

|  |  |  |  |
| --- | --- | --- | --- |
| *Экранные формы основные* | *дополнительные* | *Реализация*  *(запрос)* | *Что здесь можно использовать из серверной части* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**СКРИПТЫ**

**Серверная часть**

**Хранимые процедуры и функции**

--------------------------------------------------------------------------

-- Добавление страны

--------------------------------------------------------------------------

GO

CREATE PROCEDURE AddCountry

@ContinentName VARCHAR(50),

@Name VARCHAR(100),

@Total\_Area INT,

@Population\_Size INT

AS

BEGIN

IF (SELECT Count(Continent\_ID) FROM Continent WHERE Name = @ContinentName) = 1

IF (SELECT Count(Country\_ID) FROM Country WHERE Name = @Name) = 0

INSERT INTO Country(Continent\_ID, Name, Total\_Area, Population\_Size)

VALUES ((SELECT Continent\_ID FROM Continent WHERE Name = @ContinentName), @Name, @Total\_Area, @Population\_Size)

ELSE

PRINT 'Страна '+@Name+' уже существует'

ELSE

PRINT 'Нет такого континент: '+@ContinentName

END;

--------------------------------------------------------------------------

-- Добавление города

--------------------------------------------------------------------------

GO

CREATE PROCEDURE AddCity

@CountryName VARCHAR(50),

@Name VARCHAR(100),

@Population\_Size INT

AS

BEGIN

IF (SELECT COUNT(Country\_ID) FROM Country WHERE Name = @CountryName) = 1

IF (SELECT COUNT(City\_ID) FROM City WHERE Name = @Name) = 0

INSERT INTO City(Country\_ID, Name, Population\_Size)

VALUES ((SELECT Country\_ID FROM Country WHERE Name = @CountryName), @Name, @Population\_Size)

ELSE

PRINT 'Город '+@Name+' уже существует'

ELSE

PRINT 'Нет такой страны: '+@CountryName

END;

--------------------------------------------------------------------------

-- Добавление языка

--------------------------------------------------------------------------

GO

CREATE PROCEDURE AddLanguage

@Name VARCHAR(100)

AS IF (SELECT COUNT(Language\_ID) FROM Language WHERE Name = @Name) = 0

INSERT INTO Language (Name) VALUES (@Name);

ELSE

PRINT 'Уже содержится: '+@Name;

--------------------------------------------------------------------------

-- Привязка языка к стране

--------------------------------------------------------------------------

GO

CREATE PROCEDURE JoinCountryLanguage

@Country VARCHAR(100),

@Language VARCHAR(100)

AS

BEGIN

IF (SELECT Count(Country\_ID) FROM Country WHERE Name = @Country) = 1

IF (SELECT COUNT(Language\_ID) FROM Language WHERE Name = @Language) = 1

INSERT INTO CountryLanguage(Country\_ID,Language\_ID)

VAlUES ((SELECT Country\_ID FROM Country WHERE Name = @Country),(SELECT Language\_ID FROM Language WHERE Name = @Language));

ELSE

PRINT 'Языка нет в базе: '+@Language;

ELSE

PRINT 'Страны нет в базе: '+@Country;

END;

--------------------------------------------------------------------------

-- Добавление типа достопримечательности

--------------------------------------------------------------------------

GO

CREATE PROCEDURE AddSightType

@Name VARCHAR(100)

AS

BEGIN

IF (SELECT COUNT(SightType\_ID) FROM Sight\_Type WHERE Name = @Name) = 0

INSERT INTO Sight\_Type (Name) VALUES (@Name);

ELSE

PRINT 'Уже содержится: '+@Name;

END;

--------------------------------------------------------------------------

-- Добавление достопримечательности

--------------------------------------------------------------------------

GO

CREATE PROCEDURE AddSight

@Name VARCHAR(100),

@CityName VARCHAR(100),

@OpeningHours VARCHAR(100),

@SightType VARCHAR(100),

@Address VARCHAR(100),

@IsUNESCO BIT,

@Discription VARCHAR(1000)

AS

BEGIN

IF (SELECT COUNT(City\_ID) FROM City WHERE Name = @CityName) = 0

BEGIN

PRINT 'Нет такого города: '+@CityName;

RETURN;

END;

IF (SELECT COUNT(SightType\_ID) FROM Sight\_Type WHERE Name = @SightType) = 0

BEGIN

PRINT 'Нет такого типа достопримечательности: '+@SightType;

RETURN;

END;

INSERT INTO Sight (Name, City\_ID, OpeningHours, SightType\_ID, Address, IsUNESCO, Description)

VALUES (@Name,(SELECT City\_ID FROM City WHERE Name = @CityName),

@OpeningHours,

(SELECT SightType\_ID FROM Sight\_Type WHERE Name = @SightType),

@Address,

@IsUNESCO,

@Discription);

END;

**Триггеры**

**Представления**

----------------------------------------------------------

-- Страна-город-достопримечательности

----------------------------------------------------------

GO

CREATE VIEW CountryCitySight(Страна, Город, Достопримечательность)

AS SELECT Country.Name, City.Name, Sight.Name FROM Country join City ON Country.Country\_ID=City.Country\_ID JOIN Sight ON City.City\_ID = Sight.City\_ID

----------------------------------------------------------

-- Страна-достопримечательности

----------------------------------------------------------

GO

CREATE VIEW CountrySight(Страна, Достопримечательность)

AS SELECT Country.Name, Sight.Name FROM Country join City ON Country.Country\_ID=City.Country\_ID JOIN Sight ON City.City\_ID = Sight.City\_ID

----------------------------------------------------------

-- Тип и название достопр.

----------------------------------------------------------

GO

CREATE VIEW SightTypeSights (Тип, Название)

AS SELECT Sight\_Type.Name, Sight.Name FROM Sight\_Type JOIN Sight ON Sight.SightType\_ID = Sight\_Type.SightType\_ID

-- Вывод

SELECT \* FROM SightTypeSights

SELECT \* FROM CountryCitySight

SELECT \* FROM CountrySight

**Клиентская часть**

(запросы для экранных форм и отчетов)

-- Входят в юнеско

SELECT \* FROM Sight WHERE Sight.IsUNESCO = 1

-- Страны и языки

SELECT Country.Name AS Country, Language.Name AS Language FROM Country

JOIN CountryLanguage ON Country.Country\_ID = CountryLanguage.Country\_ID

JOIN Language ON CountryLanguage.Language\_ID = Language.Language\_ID

GROUP BY Country.Name, Language.Name

**Создание и заполнение базы данных**

CREATE DATABASE Regional\_geography\_db;

GO

USE Regional\_geography\_db;

------------------------------------------------------

--Создание таблиц и PK, FK

------------------------------------------------------

CREATE TABLE Continent (

Continent\_ID INT IDENTITY NOT NULL,

Name VARCHAR(50) UNIQUE NOT NULL,

CONSTRAINT Continent\_PK PRIMARY KEY(Continent\_ID)

)

CREATE TABLE Country (

Country\_ID INT IDENTITY NOT NULL,

Continent\_ID INT NOT NULL,

Name VARCHAR(100) UNIQUE NOT NULL,

Total\_Area INT NOT NULL,

Population\_Size INT NOT NULL,

CONSTRAINT Country\_PK PRIMARY KEY (Country\_ID),

FOREIGN KEY (Continent\_ID) REFERENCES Continent (Continent\_ID) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE Language(

Language\_ID INT IDENTITY NOT NULL,

Name VARCHAR(50) NOT NULL,

CONSTRAINT Language\_FK PRIMARY KEY (Language\_ID)

)

CREATE TABLE CountryLanguage(

ID INT IDENTITY NOT NULL,

Country\_ID INT NOT NULL,

Language\_ID INT NOT NULL,

CONSTRAINT CountryLanguage\_PK PRIMARY KEY (ID),

FOREIGN KEY (Country\_ID) REFERENCES Country (Country\_ID) ON DELETE CASCADE,

FOREIGN KEY (Language\_ID) REFERENCES Language (Language\_ID) ON DELETE CASCADE

)

CREATE TABLE City (

Country\_ID INT NOT NULL,

City\_ID INT IDENTITY NOT NULL,

Name VARCHAR(100) NOT NULL,

Population\_Size INT NOT NULL,

CONSTRAINT City\_PK PRIMARY KEY (City\_ID),

FOREIGN KEY (Country\_ID) REFERENCES Country (Country\_ID) ON DELETE CASCADE ON UPDATE CASCADE

)

CREATE TABLE Sight\_Type (

SightType\_ID INT IDENTITY NOT NULL,

Name VARCHAR(100) NOT NULL,

CONSTRAINT SightType\_PK PRIMARY KEY (SightType\_ID)

)

CREATE TABLE Sight (

Sight\_ID INT IDENTITY NOT NULL,

Name VARCHAR(100) NOT NULL,

City\_ID INT NOT NULL,

OpeningHours VARCHAR(100) NOT NULL,

SightType\_ID INT,

Address VARCHAR(200) NOT NULL,

IsUNESCO BIT,

Description VARCHAR(1000),

CONSTRAINT Sight\_PK PRIMARY KEY (Sight\_ID),

FOREIGN KEY (City\_ID) REFERENCES City (City\_ID) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (SightType\_ID) REFERENCES Sight\_Type (SightType\_ID) ON UPDATE CASCADE

)

--------------------------------------------------------------------------

-- Заполнение таблиц

--------------------------------------------------------------------------

INSERT INTO Continent(Name) VALUES ('Европа')

INSERT INTO Continent(Name) VALUES ('Азия')

INSERT INTO Continent(Name) VALUES ('Евразия')

INSERT INTO Continent(Name) VALUES ('Антарктида')

INSERT INTO Continent(Name) VALUES ('Австралия')

INSERT INTO Continent(Name) VALUES ('Южная Америка')

INSERT INTO Continent(Name) VALUES ('Северная Америка')

INSERT INTO Continent(Name) VALUES ('Африка')

-------------------------------------------------------------------------

-- Добавление данных с помощью процедур

-------------------------------------------------------------------------

GO

EXEC AddCountry 'Евразия', 'Россия', 17100000, 146781095

EXEC AddLanguage 'Русский язык'

EXEC JoinCountryLanguage 'Россия','Русский язык'

EXEC AddCity 'Россия', 'Москва', 11920000

EXEC AddCity 'Россия','Санкт-Петербург', 5383968

EXEC AddSightType 'Театр'

EXEC AddSightType 'Музей'

EXEC AddSightType 'Фонтан'

EXEC AddSightType 'Гора'

EXEC AddSightType 'Озеро'

EXEC AddSight 'Мариинский театр', 'Санкт-Петербург', '11:00–19:00', 'Театр', 'Театральная пл., 1, Санкт-Петербург, 190000', 0, 'Театр оперы и балета в Санкт-Петербурге, один из ведущих музыкальных театров России и мира.'

----------------------------------------------------------

-- Удаление представлений

----------------------------------------------------------

--DROP VIEW SightTypeSights;

--DROP VIEW CountryCitySight;

--DROP VIEW CountrySight;

----------------------------------------------------------

-- Удаление таблиц

----------------------------------------------------------

--DROP TABLE Continent;

--DROP TABLE Language;

--DROP TABLE CountryLanguage;

--DROP TABLE Country;

--DROP TABLE City;

--DROP TABLE Sight\_Type;

--DROP TABLE Sight;

--