МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

дисциплина «Разработка приложений баз данных для информационных систем»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

«Разработка серверной части информационной системы

в СУБД MS SQL Server»

Вариант №11

Выполнил:

студент группы ИТП-31, Ларчик К.И.

Принял:

доцент Асенчик О.Д.

Гомель 2020

**1. Цель работы:**

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

**2. Ход работы и результаты.**

Листинг кода на языке *Transact SQL* для создания базы данных и всех ее таблиц:

USE master

CREATE DATABASE TvChannel;

GO

ALTER DATABASE TvChannel SET RECOVERY SIMPLE

GO

USE TvChannel

CREATE TABLE Genres

(

GenreId INT PRIMARY KEY IDENTITY(1,1),

GenreName VARCHAR(16),

GenreDescription VARCHAR(32)

)

CREATE TABLE Shows

(

ShowId INT PRIMARY KEY IDENTITY(1,1),

Name VARCHAR(32),

ReleaseDate DATE,

Duration TIME,

Mark INT,

MarkMonth INT,

MarkYear INT,

GenreId INT,

Description VARCHAR(32)

)

CREATE TABLE Appeals

(

AppealId INT PRIMARY KEY IDENTITY(1,1),

FullName VARCHAR(64),

Organization VARCHAR(64),

ShowId INT,

GoalRequest VARCHAR(128)

)

CREATE TABLE Timetables

(

TimetableId INT PRIMARY KEY IDENTITY(1,1),

DayOfWeek INT,

Month INT,

Year INT,

ShowId INT,

StartTime TIME,

EndTime TIME,

StaffId INT

)

CREATE TABLE Staff

(

StaffId INT PRIMARY KEY IDENTITY(1,1),

FullName VARCHAR(64),

PositionId INT

)

CREATE TABLE Positions

(

PositionId INT PRIMARY KEY IDENTITY(1,1),

Name VARCHAR(16)

)

ALTER TABLE dbo.Shows WITH CHECK ADD CONSTRAINT FK\_Shows\_Genres FOREIGN KEY(GenreId)

REFERENCES dbo.Genres (GenreId)

GO

ALTER TABLE dbo.Appeals WITH CHECK ADD CONSTRAINT FK\_Appeals\_Shows FOREIGN KEY(ShowId)

REFERENCES dbo.Shows (ShowId)

GO

ALTER TABLE dbo.Timetables WITH CHECK ADD CONSTRAINT FK\_Timetables\_Shows FOREIGN KEY(ShowId)

REFERENCES dbo.Shows (ShowId)

GO

ALTER TABLE dbo.Timetables WITH CHECK ADD CONSTRAINT FK\_Timetables\_Staff FOREIGN KEY(StaffId)

REFERENCES dbo.Staff (StaffId)

GO

ALTER TABLE dbo.Staff WITH CHECK ADD CONSTRAINT FK\_Staff\_Positions FOREIGN KEY(PositionId)

REFERENCES dbo.Positions (PositionId)

GO

ALTER TABLE dbo.Timetables WITH CHECK ADD CONSTRAINT Check\_EndTime CHECK(EndTime > StartTime)

Код на языке *Transact SQL* для генерации тестового набора данных не менее, чем для трех таблиц:

USE TvChannel

DECLARE @Letters CHAR(52) = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz',

@i int,

@Position int,

@RowCount int,

@RowIndex int,

@MinLetters int,

@MaxLetters int,

@LettersLimit int,

-- Таблица "Genres".

@GenreName varchar(16),

@GenreDescription varchar(32),

-- Таблица "Shows".

@ShowName varchar(32),

@ShowReleaseDate date,

@ShowDuration time,

@ShowMark int,

@ShowMarkMonth int,

@ShowMarkYear int,

@ShowGenreId int,

@ShowDescription varchar(32),

-- Таблица "Timetables".

@DayOfWeek int,

@Month int,

@Year int,

@ShowId int,

@StartTime time,

@EndTime time

SET NOCOUNT ON

-- Таблица "Genres".

SET @RowCount = 500

SET @RowIndex = 1

SET @MinLetters = 8

WHILE @RowIndex <= @RowCount

BEGIN

-- Название жанра.

SET @MaxLetters = 16

SET @LettersLimit = @MinLetters + RAND()\*(@MaxLetters - @MinLetters)

SET @i = 1

SET @GenreName = ''

WHILE @i <= @LettersLimit

BEGIN

SET @Position = RAND()\*52

SET @GenreName = @GenreName + SUBSTRING(@Letters, @Position, 1)

SET @i += 1

END

-- Описания жанра.

SET @MaxLetters = 32

SET @LettersLimit = @MinLetters + RAND()\*(@MaxLetters - @MinLetters)

SET @i = 1

SET @GenreDescription = ''

WHILE @i <= @LettersLimit

BEGIN

SET @Position = RAND()\*52

SET @GenreDescription = @GenreDescription + SUBSTRING(@Letters, @Position, 1)

SET @i += 1

END

INSERT INTO Genres (GenreName, GenreDescription) VALUES (@GenreName, @GenreDescription)

SET @RowIndex += 1

END

-- Таблица "Shows".

SET @RowCount = 20000;

SET @RowIndex = 1;

WHILE @RowIndex <= @RowCount

BEGIN

-- Название телепередачи.

SET @MinLetters = 8;

SET @MaxLetters = 32;

SET @LettersLimit = @MinLetters + RAND()\*(@MaxLetters - @MinLetters)

SET @ShowName = ''

SET @i = 1

WHILE @i <= @LettersLimit

BEGIN

SET @Position = RAND()\*52

SET @ShowName = @ShowName + SUBSTRING(@Letters, @Position, 1)

SET @i += 1

END

-- Дата выхода.

SET @ShowReleaseDate = dateadd(DAY, -RAND()\*15000, GETDATE())

-- Продолжительность

DECLARE @TempDuration int = RAND()\*(10801 - 3600) + 3600

SET @ShowDuration = CONVERT(VARCHAR(8), (Select DATEADD(SECOND,DATEDIFF(DAY,0, @TempDuration),'00:00:00')), 108)

-- Оценка

SET @ShowMark = RAND()\*(11-1)+1

SET @ShowMarkMonth = RAND()\*(13-1)+1

SET @ShowMarkYear = RAND()\*(2021-2010)+2010

-- Внешний ключ к т. "Genres".

SET @ShowGenreId = RAND()\*(501-1)+1

-- Описания телепередачи.

SET @MaxLetters = 32

SET @LettersLimit = @MinLetters + RAND()\*(@MaxLetters - @MinLetters)

SET @i = 1

SET @ShowDescription = ''

WHILE @i <= @LettersLimit

BEGIN

SET @Position = RAND()\*52

SET @ShowDescription = @ShowDescription + SUBSTRING(@Letters, @Position, 1)

SET @i += 1

END

INSERT INTO Shows (Name, ReleaseDate, Duration, Mark, MarkMonth, MarkYear, GenreId, Description)

VALUES (@ShowName, @ShowReleaseDate, @ShowDuration, @ShowMark, @ShowMarkMonth, @ShowMarkYear, @ShowGenreId, @ShowDescription)

SET @RowIndex += 1

END

-- Таблица "Timetables".

SET @RowCount = 20000

SET @RowIndex = 1

WHILE @RowIndex <= @RowCount

BEGIN

SET @DayOfWeek = RAND()\*(8-1)+1

SET @Month = RAND()\*(13-1)+1

SET @Year = RAND()\*(2021-2010)+2010

SET @ShowId = RAND()\*(20001 - 1) + 1

DECLARE @TempStartTime int = RAND()\*(75600 - 28800) + 28800

SET @StartTime = CONVERT(VARCHAR(8), (Select DATEADD(SECOND,DATEDIFF(DAY,0, @TempStartTime),'00:00:00')), 108)

SET @EndTime = CONVERT(VARCHAR(8), (SELECT DATEADD(SECOND, DATEDIFF(SECOND,0,(SELECT Shows.Duration FROM Shows WHERE @ShowId = Shows.ShowId)), @StartTime)), 108)

INSERT INTO Timetables (DayOfWeek, Month, Year, ShowId, StartTime, EndTime) VALUES

(@DayOfWeek, @Month, @Year, @ShowId, @StartTime, @EndTime)

SET @RowIndex += 1

END

Листинг кода на языке *Transact SQL* не менее трех представлений:

USE TvChannel

GO

CREATE VIEW [dbo].[View\_Shows]

AS

SELECT dbo.Shows.Name, dbo.Shows.ReleaseDate, dbo.Genres.GenreName,

dbo.Shows.Description, dbo.Shows.Duration, dbo.Shows.Mark, dbo.Shows.MarkMonth,

dbo.Shows.MarkYear

FROM dbo.Shows JOIN dbo.Genres on dbo.Shows.GenreId = dbo.Genres.GenreId

GO

CREATE VIEW [dbo].[View\_Timetables]

AS

SELECT dbo.Shows.Name, dbo.Genres.GenreName, dbo.Timetables.DayOfWeek, dbo.Timetables.Month, dbo.Timetables.Year,

dbo.Timetables.StartTime, dbo.Timetables.EndTime

FROM dbo.Shows JOIN dbo.Genres on dbo.Shows.GenreId = dbo.Genres.GenreId

JOIN dbo.Timetables on dbo.Shows.ShowId = dbo.Timetables.ShowId

GO

CREATE VIEW [dbo].[View\_Genres]

AS

SELECT dbo.Genres.GenreName, dbo.Genres.GenreDescription

FROM dbo.Genres

Листинг кода на языке *Transact SQL* не менее трех хранимых процедур:

use TvChannel

go

create procedure sp\_InsertShow

@Name varchar(32),

@ReleaseDate datetime,

@Duration time,

@Mark int,

@MarkMonth int,

@MarkYear int,

@GenreName varchar(16),

@Description varchar(32)

as

declare @TempDate datetime = dateadd(month, @MarkMonth, @ReleaseDate)

set @TempDate = dateadd(year, @MarkYear, @ReleaseDate)

if @ReleaseDate > @TempDate

begin

set @Mark = null

set @MarkMonth = null

set @MarkYear = null

end

declare @GenreId int = (select dbo.Genres.GenreId from dbo.Genres where dbo.Genres.GenreName = @GenreName)

insert into dbo.Shows (Name, ReleaseDate, Duration, Mark, MarkMonth, MarkYear, GenreId, Description)

select @Name, @ReleaseDate, @Duration, @Mark, @MarkMonth, @MarkYear, @GenreId, @Description

go

create procedure sp\_UpdateShow

@Name varchar(32),

@ReleaseDate datetime,

@Duration time,

@Mark int,

@MarkMonth int,

@MarkYear int,

@GenreName varchar(16),

@Description varchar(32),

@ShowId int

as

declare @TempDate datetime = dateadd(month, @MarkMonth, @ReleaseDate)

set @TempDate = dateadd(year, @MarkYear, @ReleaseDate)

if @ReleaseDate > @TempDate

begin

set @Mark = null

set @MarkMonth = null

set @MarkYear = null

end

declare @GenreId int = (select dbo.Genres.GenreId from dbo.Genres where dbo.Genres.GenreName = @GenreName)

update dbo.Shows

set dbo.Shows.Name = @Name,

dbo.Shows.ReleaseDate = @ReleaseDate,

dbo.Shows.Duration = @Duration,

dbo.Shows.Mark = @Mark,

dbo.Shows.MarkMonth = @MarkMonth,

dbo.Shows.MarkYear = @MarkYear,

dbo.Shows.GenreId = @GenreId,

dbo.Shows.Description = @Description

where dbo.Shows.ShowId = @ShowId

go

create procedure sp\_InsertTimetables

@DayOfWeek int,

@Month int,

@Year int,

@ShowName varchar(32),

@StartTime time,

@StaffId int null

as

declare @ShowId int = (select dbo.Shows.ShowId from dbo.Shows where dbo.Shows.Name = @ShowName)

declare @EndTime datetime = CONVERT(VARCHAR(8), (SELECT DATEADD(SECOND, DATEDIFF(SECOND,0,(SELECT Shows.Duration FROM Shows WHERE @ShowId = Shows.ShowId)), @StartTime)), 108)

insert into dbo.Timetables (DayOfWeek, Month, Year, ShowId, StartTime, EndTime)

select @DayOfWeek, @Month, @Year, @ShowId, @StartTime, @EndTime

go

**3. Вывод:** в результате выполнения данной лабораторной работы были созданы база данных, таблица базы данных, также таблицы были заполнены тестовыми наборами данных, были созданы представления для удобного отображения данных и были написаны хранимые процедуры для вставки данных в три таблицы.