**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

группы ИТП-31

Шух Н. С.

Проверил: преподаватель

Ястребов А.А.

Гомель 2023

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

**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.

На рисунке 1 представлен вариант задания.

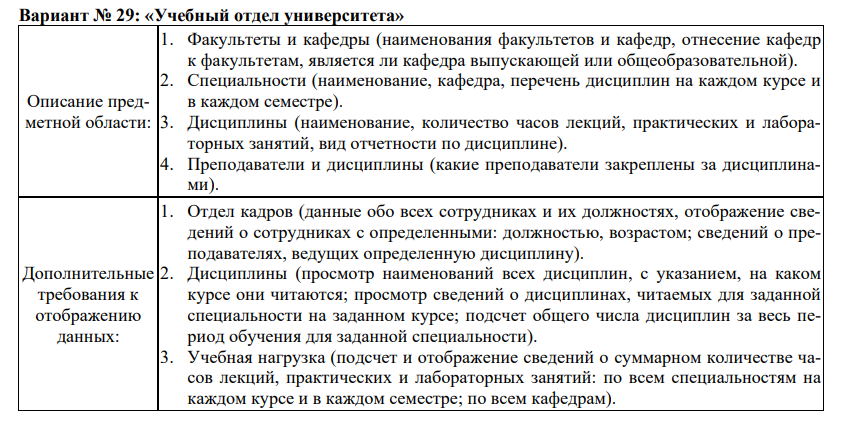


Рисунок 1 – вариант задания

**Ход работы**

Диаграмма созданной базы данных представлена на рисунке 2.

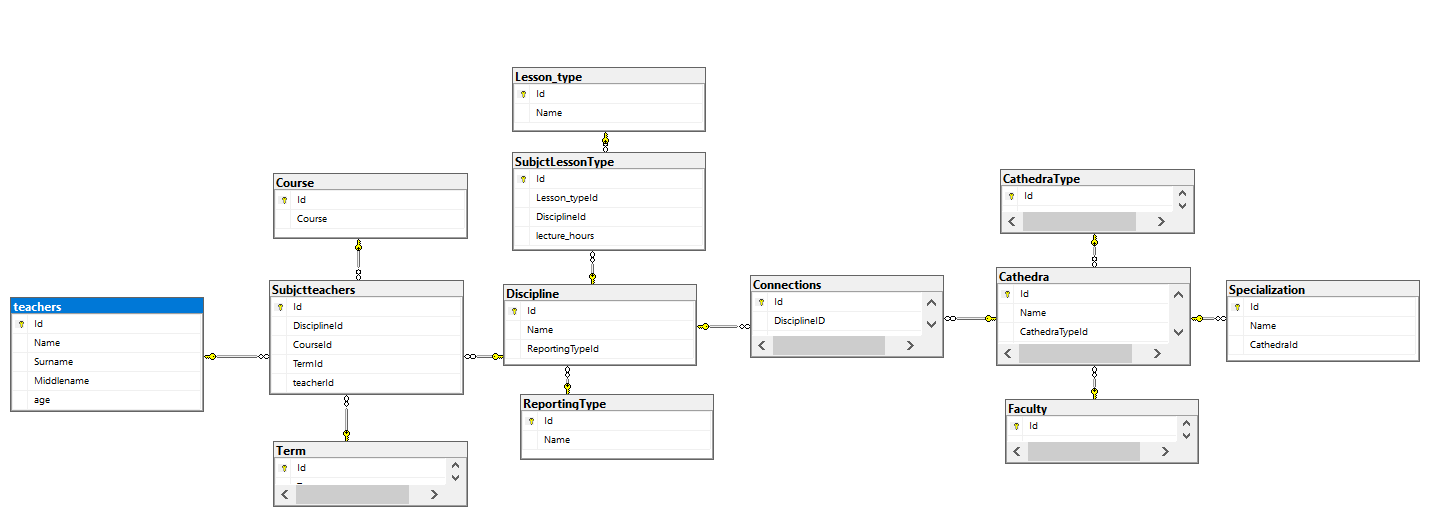


Рисунок 2 – Диаграмма базы данных

C помощью средств *MS SQl* был создан *Initializer* для заполнения таблиц, представленный на рисунке 3.



Рисунок 3 – *Initializer* для заполнения таблиц

Текст представлений и процедур представлен в приложении А.

**Вывод**: в ходе лабораторной работы была разработана, серверная часть клиентно-серверной информационной системы, основанной на базе данных в заданной предметной области средствами СУБД *MS SQL* *Server*. .

**ПРИЛОЖЕНИЕ А**

(обязательное)

Запрос на создание базы данных:

USE master

IF DB\_ID('LR1') IS NOT NULL

DROP DATABASE LR1

USE master

CREATE DATABASE LR1

Go

USE LR1

/\*Типы кафедры\*/

CREATE TABLE CathedraType(

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

Name NVARCHAR(150),

);

/\*Факультет\*/

CREATE TABLE Faculty(

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

Name NVARCHAR(150),

);

/\*Кафедра\*/

CREATE TABLE Cathedra(

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

Name NVARCHAR(150),

CathedraTypeId INT NOT NULL,

FacultyId INT NOT NULL,

FOREIGN KEY (CathedraTypeId) REFERENCES CathedraType(Id) ON DELETE CASCADE,

FOREIGN KEY (FacultyId) REFERENCES Faculty(Id) ON DELETE CASCADE,

);

/\*Специальность\*/

CREATE TABLE Specialization(

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

Name NVARCHAR(150),

CathedraId INT NOT NULL,

FOREIGN KEY (CathedraId) REFERENCES Cathedra(Id) ON DELETE CASCADE,

);

/\*Вид отчетности\*/

CREATE TABLE ReportingType(

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

Name NVARCHAR(150),

);

/\*Дисцыплина\*/

CREATE TABLE Discipline(

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

Name NVARCHAR(150),

ReportingTypeId INT NOT NULL,

FOREIGN KEY (ReportingTypeId) REFERENCES ReportingType(Id) ON DELETE CASCADE,

);

/\*Тип занятий\*/

CREATE TABLE Lesson\_type(

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

Name NVARCHAR(150),

);

/\*Subjct Lesson Type\*/

CREATE TABLE SubjctLessonType(

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

Lesson\_typeId INT NOT NULL,

DisciplineId INT NOT NULL,

lecture\_hours INT NOT NULL,

FOREIGN KEY (DisciplineId) REFERENCES Discipline(Id) ON DELETE CASCADE,

FOREIGN KEY (Lesson\_typeId) REFERENCES Lesson\_type(Id) ON DELETE CASCADE,

);

/\*преподаватели\*/

CREATE TABLE teachers(

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

Name NVARCHAR(150),

Surname NVARCHAR(150),

Middlename NVARCHAR(150),

age INT NOT NULL,

);

/\*семестр\*/

CREATE TABLE Term(

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

Term INT NOT NULL,

);

/\*Курс\*/

CREATE TABLE Course(

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

Course INT NOT NULL,

);

/\*Subjctteachers\*/

CREATE TABLE Subjctteachers(

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

DisciplineId INT NOT NULL,

CourseId INT NOT NULL,

TermId INT NOT NULL,

teacherId INT NOT NULL,

FOREIGN KEY (DisciplineId) REFERENCES Discipline(Id) ON DELETE CASCADE,

FOREIGN KEY (teacherId) REFERENCES teachers(Id) ON DELETE CASCADE,

FOREIGN KEY (CourseId) REFERENCES Course(Id) ON DELETE CASCADE,

FOREIGN KEY (TermId) REFERENCES Term(Id) ON DELETE CASCADE,

);

/\*Связь\*/

CREATE TABLE Сonnections(

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

DisciplineID INT NOT NULL,

CathedraID INT NOT NULL,

FOREIGN KEY (DisciplineID) REFERENCES Discipline(Id) ON DELETE CASCADE,

FOREIGN KEY (CathedraID) REFERENCES Cathedra(Id) ON DELETE CASCADE,

);

Листинг Table\_Cathedra.cs:

use LR1

go

Declare @Name char(50),

@i int,

@RowCount int,

@CathedraTypeId int,

@FacultyId int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'Cathedra1'

When @Name = 2 Then 'Cathedra2'

When @Name = 3 Then 'Cathedra3'

When @Name = 4 Then 'Cathedra4'

When @Name = 5 Then 'Cathedra5'

When @Name = 6 Then 'Cathedra6'

When @Name = 7 Then 'Cathedra7'

When @Name = 8 Then 'Cathedra8'

When @Name = 9 Then 'Cathedra9'

When @Name = 10 Then 'Cathedra10'

Else 'Cathedra11'

End;

Set @CathedraTypeId = FLOOR(Rand()\*1000)

Set @FacultyId = FLOOR(Rand()\*1000)

Insert into dbo.Cathedra(Name,CathedraTypeId,FacultyId) values(@Name,@CathedraTypeId,@FacultyId)

Set @i += 1

End

Листинг Table\_ CathedraType.cs:

use LR1

go

Declare @Name char(20),

@i int,

@RowCount int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'CathedraType1'

When @Name = 2 Then 'CathedraType2'

When @Name = 3 Then 'CathedraType3'

When @Name = 4 Then 'CathedraType4'

When @Name = 5 Then 'CathedraType5'

When @Name = 6 Then 'CathedraType6'

When @Name = 7 Then 'CathedraType7'

When @Name = 8 Then 'CathedraType8'

When @Name = 9 Then 'CathedraType9'

When @Name = 10 Then 'CathedraType10'

Else 'CathedraType11'

End;

Insert into dbo.CathedraType(Name) values(@Name)

Set @i += 1

End

Листинг Table\_ Course.cs:

use LR1

go

Declare @Course int,

@i int,

@RowCount int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Course = Round(Rand()\*7, 0)

Select @Course =

Case

When @Course = 1 Then 1

When @Course = 2 Then 2

When @Course = 3 Then 3

When @Course = 4 Then 4

When @Course = 5 Then 5

When @Course = 6 Then 6

When @Course = 7 Then 7

Else 8

End;

Insert into dbo.Course(Course) values(@Course)

Set @i += 1

End

Листинг Table\_ Discipline.cs:

use LR1

go

Declare @Name char(50),

@i int,

@RowCount int,

@ReportingTypeId int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'Discipline1'

When @Name = 2 Then 'Discipline2'

When @Name = 3 Then 'Discipline3'

When @Name = 4 Then 'Discipline4'

When @Name = 5 Then 'Discipline5'

When @Name = 6 Then 'Discipline6'

When @Name = 7 Then 'Discipline7'

When @Name = 8 Then 'Discipline8'

When @Name = 9 Then 'Discipline9'

When @Name = 10 Then 'Discipline10'

Else 'Discipline11'

End;

Set @ReportingTypeId= FLOOR(Rand()\*1000)

Insert into dbo.Discipline(Name,ReportingTypeId) values(@Name,@ReportingTypeId)

Set @i += 1

End

Листинг Table\_ Faculty.cs:

use LR1

go

Declare @Name char(20),

@i int,

@RowCount int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'Faculty1'

When @Name = 2 Then 'Faculty2'

When @Name = 3 Then 'Faculty3'

When @Name = 4 Then 'Faculty4'

When @Name = 5 Then 'Faculty5'

When @Name = 6 Then 'Faculty6'

When @Name = 7 Then 'Faculty7'

When @Name = 8 Then 'Faculty8'

When @Name = 9 Then 'Faculty9'

When @Name = 10 Then 'Faculty10'

Else 'Faculty11'

End;

Insert into dbo.Faculty(Name) values(@Name)

Set @i += 1

End

Листинг Table\_ Lesson\_type.cs:

use LR1

go

Declare @Name char(50),

@i int,

@RowCount int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'Lesson\_type1'

When @Name = 2 Then 'Lesson\_type2'

When @Name = 3 Then 'Lesson\_type3'

When @Name = 4 Then 'Lesson\_type4'

When @Name = 5 Then 'Lesson\_type5'

When @Name = 6 Then 'Lesson\_type6'

When @Name = 7 Then 'Lesson\_type7'

When @Name = 8 Then 'Lesson\_type8'

When @Name = 9 Then 'Lesson\_type9'

When @Name = 10 Then 'Lesson\_type10'

Else 'Lesson\_type11'

End;

Insert into dbo.Lesson\_type(Name) values(@Name)

Set @i += 1

End

Листинг Table\_ ReportingType.cs:

use LR1

go

Declare @Name char(50),

@i int,

@RowCount int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'ReportingType1'

When @Name = 2 Then 'ReportingType2'

When @Name = 3 Then 'ReportingType3'

When @Name = 4 Then 'ReportingType4'

When @Name = 5 Then 'ReportingType5'

When @Name = 6 Then 'ReportingType6'

When @Name = 7 Then 'ReportingType7'

When @Name = 8 Then 'ReportingType8'

When @Name = 9 Then 'ReportingType9'

When @Name = 10 Then 'ReportingType10'

Else 'ReportingType11'

End;

Insert into dbo.ReportingType(Name) values(@Name)

Set @i += 1

End

Листинг Table\_ Specialization.cs:

use LR1

go

Declare @Name char(50),

@i int,

@RowCount int,

@CathedraId int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'Specialization1'

When @Name = 2 Then 'Specialization2'

When @Name = 3 Then 'Specialization3'

When @Name = 4 Then 'Specialization4'

When @Name = 5 Then 'Specialization5'

When @Name = 6 Then 'Specialization6'

When @Name = 7 Then 'Specialization7'

When @Name = 8 Then 'Specialization8'

When @Name = 9 Then 'Specialization9'

When @Name = 10 Then 'Specialization10'

Else 'Specialization11'

End;

Set @CathedraId= FLOOR(Rand()\*1000)

Insert into dbo.Specialization(Name,CathedraId) values(@Name,@CathedraId)

Set @i += 1

End

Листинг Table\_ SubjctLessonTyp.cs:

use LR1

go

Declare @lecture\_hours int,

@i int,

@RowCount int,

@Lesson\_typeId INT,

@DisciplineId INT

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @lecture\_hours = Round(Rand()\*10, 0)

Select @lecture\_hours =

Case

When @lecture\_hours = 1 Then 50

When @lecture\_hours = 2 Then 60

When @lecture\_hours = 3 Then 70

When @lecture\_hours = 4 Then 80

When @lecture\_hours = 5 Then 90

When @lecture\_hours = 6 Then 100

When @lecture\_hours = 7 Then 110

When @lecture\_hours = 8 Then 95

When @lecture\_hours = 9 Then 97

When @lecture\_hours = 10 Then 66

Else 88

End;

Set @Lesson\_typeId = FLOOR(Rand()\*1000)

Set @DisciplineId = FLOOR(Rand()\*1000)

Insert into dbo.SubjctLessonType(lecture\_hours,Lesson\_typeId,DisciplineId) values(@lecture\_hours,@Lesson\_typeId,@DisciplineId)

Set @i += 1

End

Листинг Table\_ Subjctteachers.cs:

use LR1

go

Declare @i int,

@RowCount int,

@DisciplineId INT,

@CourseId INT,

@TermId INT,

@teacherId INT

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @DisciplineId = FLOOR(Rand()\*1000)

Set @CourseId = FLOOR(Rand()\*1000)

Set @TermId = FLOOR(Rand()\*1000)

Set @teacherId = FLOOR(Rand()\*1000)

Insert into dbo.Subjctteachers(DisciplineId,CourseId,TermId,teacherId) values(@DisciplineId,@CourseId,@TermId,@teacherId)

Set @i += 1

End

Листинг Table\_ teachers.cs:

use LR1

go

Declare @Name char(50),

@Surname char(50),

@Middlename char(50),

@i int,

@Age int,

@RowCount int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Name = Round(Rand()\*10, 0)

Select @Name =

Case

When @Name = 1 Then 'Никита'

When @Name = 2 Then 'Максим'

When @Name = 3 Then 'Александр'

When @Name = 4 Then 'Артем'

When @Name = 5 Then 'Илья'

When @Name = 6 Then 'Егор'

When @Name = 7 Then 'Иван'

When @Name = 8 Then 'Елена'

When @Name = 9 Then 'Светлана'

When @Name = 10 Then 'Кристина'

Else 'Сергей'

End;

Set @Surname = Round(Rand()\*10, 0)

Select @Surname =

Case

When @Surname = 1 Then 'Шух'

When @Surname = 2 Then 'Ялченка'

When @Surname = 3 Then 'Рябый'

When @Surname = 4 Then 'Бондарев'

When @Surname = 5 Then 'Волчек'

When @Surname = 6 Then 'Маслак'

When @Surname = 7 Then 'Щутко'

When @Surname = 8 Then 'Король'

When @Surname = 9 Then 'Голец'

When @Surname = 10 Then 'Данилков'

Else 'Ковтун'

End;

Set @Middlename = Round(Rand()\*10, 0)

Select @Middlename =

Case

When @Middlename = 1 Then 'Юрьевич'

When @Middlename = 2 Then 'Сергеевич'

When @Middlename = 3 Then 'Геннадьевич'

When @Middlename = 4 Then 'Алексеевич'

When @Middlename = 5 Then 'Викторович'

When @Middlename = 6 Then 'Николаевич'

When @Middlename = 7 Then 'Дмитриевич'

When @Middlename = 8 Then 'Леонидович'

When @Middlename = 9 Then 'Игорович'

When @Middlename = 10 Then 'Витальевич'

Else 'Джозеф'

End;

Set @Age = FLOOR(Rand()\*50)

Insert into dbo.teachers(Name,Surname,Middlename,Age) values(@Name,@Surname,@Middlename,@Age)

Set @i += 1

End

Листинг Table\_ Term.cs:

use LR1

go

Declare @Term int,

@i int,

@RowCount int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @Term = Round(Rand()\*3, 0)

Select @Term =

Case

When @Term = 1 Then 1

When @Term = 2 Then 2

When @Term = 3 Then 3

Else 4

End;

Insert into dbo.Term(Term) values(@Term)

Set @i += 1

End

Листинг Table\_ Сonnections.cs:

use LR1

go

Declare @i int,

@RowCount int,

@DisciplineID int,

@CathedraID int

Set @RowCount = 1000

Set @i = 1

While @i <= @RowCount

Begin

Set @DisciplineID = FLOOR(Rand()\*1000)

Set @CathedraID = FLOOR(Rand()\*1000)

Insert into dbo.Сonnections(DisciplineID,CathedraID) values(@DisciplineID,@CathedraID)

Set @i += 1

End

Код представления:

use LR1

go

Alter VIEW LessonType\_v

AS SELECT

SLT.Id as LessonTypeId,

LT.Name as LessonType,

D.Name as Discipline,

SLT.lecture\_hours as lecture\_hours

From SubjctLessonType as SLT

join Lesson\_type as LT on LT.Id = SLT.id

join Discipline as D on D.Id = SLT.Id

go

Alter VIEW Cathedra\_v

AS SELECT

C.Id as CathedraTypeId,

CT.Name as CathedraType,

F.Name as Faculty

From Cathedra as C

join CathedraType as CT on CT.Id = C.id

join Faculty as F on F.Id = C.Id

go

create VIEW Subjctteachers\_v

AS SELECT

ST.Id as CourseId,

C.Course as Course,

D.Name as Discipline,

TE.Name as teachers

From Subjctteachers as ST

join Course as C on C.Id = ST.Id

join Term as T on T.Id = ST.Id

join teachers as TE on TE.Id = ST.Id

join Discipline as D on D.Id = ST.Id

go

Select \* From Cathedra\_v

Select \* From Subjctteachers\_v

Select \* From LessonType\_v

Код процедуры 1

use LR1

go

create procedure AddCathedra

@name char(20),

@CathedraTypeId char(20),

@FacultyId int

as

insert into Cathedra(Name, CathedraTypeId, FacultyId)

values(@name, @CathedraTypeId, @FacultyId)

Код процедуры 2:

use LR1

go

create procedure DelCathedra

@id int

as

delete from Cathedra

where Id = @id

Код процедуры 3:

use LR1

go

create procedure UpdateCathedra

@id int,

@name char(20),

@CathedraTypeId char(20),

@FacultyId int

as

update Cathedra

set Name = @name, CathedraTypeId = @CathedraTypeId, FacultyId = @FacultyId

where Id = @id