

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Балагурин Иван Леонидович

Должность: И.о. ректора

Дата подписания: 19.08.2023 03:37:58

Уникальный программный ключ:

2a04bb882d7e0b2e2a11e0c4a0c5e41f

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
ПРИКЛАДНОЙ ИНФОРМАТИКИ В ЭКОНОМИКЕ  
(ИТиПИВЭ)**

**УЧЕБНО-МЕТОДИЧЕСКИЕ  
УКАЗАНИЯ**

**к выполнению лабораторных работ по дисциплине «Информационные системы и технологии» для студентов направления подготовки бакалавров 09.03.03 – «Прикладная информатика», профиль «Прикладная информатика в экономике». Часть 1.**

**Махачкала, 2020 г.**

**УДК 37.01**

Учебно-методические указания к выполнению лабораторных работ по дисциплине «Информационные системы и технологии» для студентов направления подготовки бакалавров 09.03.03– «Прикладная информатика», профиль «Прикладная информатика в экономике». Часть 1.-Махачкала, ДГТУ, 2020, 36 с.

Учебно-методические указания предназначены для студентов дневной и заочной форм обучения, занимающихся по профилю подготовки бакалавров «Прикладная информатика в экономике».

Методические указания содержат краткие теоретические сведения, необходимые для выполнения лабораторных работ по переводу чисел из одной системы счисления в другую, соответствующие блок-схемы алгоритмов перевода чисел, начальные сведения по использованию СУБД MS SQL Server для создания баз данных. Кроме того, в методических указаниях приведены индивидуальные задания к лабораторным работам.

Составитель:

Абдулгалимов А.М., зав. кафедрой ИТиПИВЭ, д.э.н.,  
профессор

Рецензенты:

Баламирзоев А.Г. - д.т.н., профессор кафедры математики и информатики, Махачкалинский филиал Московского автомобильно-дорожного государственного технического университета, г. Махачкала;

Джабраилов Х.С., директор ООО «ИВЦ «Сигма»,  
к.э.н., г. Махачкала

Печатается по решению Совета Дагестанского государственного технического университета от «\_\_\_\_\_» \_\_\_\_\_ 2020 г.

## Введение

Настоящие учебно-методические указания подготовлены в соответствии с Федеральным государственным образовательным стандартом по направлению подготовки (специальности) 09.03.03-«Прикладная информатика» и уровню высшего образования «Бакалавриат», утвержденный приказом Минобрнауки России от 19.09.2017 № 922; Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры, утвержденный приказом Минобрнауки России от 5 апреля 2017 года № 301; Уставом Дагестанского государственного технического университета; Основной профессиональной образовательной программой (ОПОП) по направлению 09.03.03 – «Прикладная информатика» (профиль подготовки бакалавров: «Прикладная информатика в экономике»).

Внедрение ЭВМ во все сферы человеческой деятельности требует от специалистов разного профиля овладения навыками использования информационных систем и основами их разработки и создания. Поэтому важной составляющей подготовки студентов по профилю - «Прикладная информатика в экономике является изучение таких направлений использования ЭВМ, как проектирование и построение информационных систем (ИС), практическое использование информационных систем для автоматизации деятельности предприятий и организаций.

В учебно-методических указаниях в кратком виде приводятся необходимые теоретические сведения из курса «Информационные системы и технологии» для выполнения лабораторных работ.

Решение задач ориентировано на использование ПЭВМ и соответствующего программного обеспечения, например, C++, MS SQL Server . Учебно-методические указания являются полезными при выполнении лабораторных работ по курсам: базы данных и проектирование информационных систем.

## 1. Лабораторная работа №1

Тема: «Системы счисления».

**Цель работы – изучение систем счисления. Перевод чисел из одной системы счисления (с/с) в другую.**

Число, как выражение количественных характеристик окружающего мира, возникло в глубокой древности. Окончательное становление теория действительных чисел получила во второй половине XIX века, в связи с потребностями математического анализа. Наряду с действительными числами, начиная с XVI века были введены комплексные числа. Свойства чисел, их теорию изучают такие специальные математические дисциплины как теория чисел, алгебра, математический анализ. Полагая, что читатель имеет общие сведения по теории чисел, отметим (без доказательства), что любое вещественное число, в десятичной системе счисления, может быть представлено с любой, наперед заданной точностью, рядом вида:

$$x = \sum_{i=-\infty}^{i=+\infty} a_i \cdot 10^i$$

Число, как мера количества, может иметь различное представление (выражение), которое связано с понятием системы счисления.

Существуют две разновидности систем счисления – непозиционные, например, римские цифры, и позиционные – к примеру, известная нам 10-я с/с. Примеры других систем счисления мы можем обнаружить в языковых конструкциях: семь дней в неделе, 24 часа в сутках, 60 минут в часе и т.д. Одно и то же количество может быть выражено различными числами, в зависимости от используемой системы счисления.

Непозиционные с/с имеют свои особенности, на которых мы останавливаться не будем.

В позиционной системе счисления каждая цифра имеет свой "вес", в зависимости от того места, которое она занимает в числе.

Если в математике, в других научных дисциплинах, мы оперируем числами в промежутке  $(-\infty, +\infty)^1$ , а также можем использовать неограниченное число знаков в дробной части числа, то в компьютере

---

<sup>1</sup> Вещественные числа обладают свойством непрерывности. См.: Кудрявцев Л.Д. Курс математического анализа (в 2-х т.). Т.1. Учебник. М., 1981. 687 с.

используется конечное число чисел с определенным минимальным дискретным шагом.<sup>2</sup> В силу сказанного, реальные числа представляются в компьютерной технике приближенно.

Исторически сложилось, что цифровые вычислительные машины (ЦВМ), формально представляющие собой материализацию машины Тьюринга, работают в двоичном коде, то есть используют 2-ичную с/с. Наибольший интерес в приложениях к вычислительной технике, ЭВМ, представляют системы счисления с основаниями: 2, 8, 16.

Значение числа в системе счисления с натуральным основанием может быть представлено следующей формулой:

$$X = \text{sign}X(\sum a_i \cdot S^i) \quad (1)$$

где:  $S^3$  – натуральное основание (для простоты положим  $S > 1$ )

$a$  – коэффициент в разложении (1), то есть значение разряда, ( $a = 0, 1, \dots, S-1$ )

$i$  – целочисленная переменная из промежутка  $(-\infty, +\infty)$

$\text{sign}X$  – знак числа ("+" - для положительных чисел, "-"

- для отрицательных)

Пример. Пусть задано целое положительное число 567 в 10-й с/с. Его разложение по основанию 10 будет иметь вид:  $567 = 7 \times 10^0 + 6 \times 10^1 + 5 \times 10^2$ .

Операции над числами<sup>4</sup>. Остановимся на двух простейших операциях сложении и умножении. Практически, чтобы осуществлять их, необходимо использовать таблицы сложения и умножения. Означенные таблицы для 10-й с/с изучаются в начальных классах средней школы. Считая известными вышеназванные операции, рассмотрим:

Перевод чисел из одной с/с в другую.

---

<sup>2</sup> См.: Погорелый С.Д., Слободянюк Т.Ф. Программное обеспечение микропроцессорных систем. // Справ. Киев, 1989. 301 с; Донован Д. Системное программирование. М., 1975. 544 с; Брич З.С., Воюш В.И., Дегтярева Г.С., Ковалевич Э.В. Программирование на языке Ассемблера ЕС ЭВМ. М., 1975. 296 с; Вирт Н. Систематическое программирование. Введение. М., 1977. 184 с; Баррон Д. Введение в языки программирования. М., 1980. 190 с. и др.

<sup>3</sup> В общем случае  $S$  может быть и не целым

<sup>4</sup> Общие сведения об операциях см.: Кострикин А.И. Введение в алгебру. М., 1977. 495 с; Кудрявцев Л.Д. Курс математического анализа (в двух томах). М., 1981. Т.1. 687 с., ил; Куратовский К. Топология. Т.1. М., 1966. 594 с. и др.

*При переводе числа  $X$  из одной с/с в другую отдельно переводится целая часть и отдельно дробная часть.*

### Перевод целого числа

Пусть  $X > 0$  и  $X_s = \sum a_i \cdot S^i$ , где  $i=0, +\infty$ . Требуется найти представление числа  $X$  в системе счисления с основанием  $R$ .

$$X_r = \sum b_q \cdot R^q \quad (2)$$

Значение (величина числа)  $X$  не зависит от конкретного выражения системы счисления (и даже свойства ее позиционности), то есть:

$$X_s = X_r \quad (3)$$

Описание алгоритма перевода.

1. Разделим обе части равенства (3) на  $R$  *по правилам деления в системе счисления с основанием  $S$* . Остаток от этого деления совпадает с цифрой  $R_0$ .
2. Отбросим в обеих частях равенства остатки от деления. Оставшиеся после этого части разложения опять приравняем друг другу и снова разделим на  $R$ .
3. Остаток от этого деления совпадает с  $R_1$ .
4. Повторяем операцию до тех пор, пока частное не окажется равным 0.

Получим все значения  $R_q$ . Количество полученных значащих цифр определит параметр  $q$ .

Пример.  $X_{10} = 234$ .

В 6 с/с = 1030

В 5 с/с = 1414

В 7 с/с = 453

Проверка правильности перевода – воспользуемся разложением по формуле (2).

При переводе целых чисел всегда получаются целые числа.

### Перевод дробного числа

В отличие от перевода целых чисел, при переводе дробных указывается точность перевода. Это связано с тем, что конечная дробь в некоторой системе счисления, не будет таковой в другой системе.

Пусть  $y$  – правильная дробь ( $<1$ ), записанная в с/с с основанием  $S$ . Требуется записать эту дробь в с/с с основанием  $R$  с заданной точностью  $\varepsilon$ . Величину точности задают в виде величины отрицательной степени основания.

Как и в случае с целыми числами, приравняем значение  $y$  в различных системах счисления:

$$y_s = y_r \quad (4)$$

Умножим  $y_s$  на  $R$  по правилам умножения чисел в  $c/s$  с основанием  $S$ . Целая часть произведения совпадает с  $V_{-1}$  – выделим дробную часть произведения и умножим снова на  $R$ . Целая часть будет представлять  $V_{-2}$ . процесс продолжается до тех пор, пока не будет определена последняя цифра  $V_{-q}$ .

Пример. Преобразовать число  $y_{10} = 0,45$  в двоичную систему счисления с  $\varepsilon = 2^{-4}$ .

$$0,45_{10} = 0,11100110011..._2$$

Системы счисления с основание 2, 8, 16 имеют особое значение в компьютерных операциях. Переход от одной к другой осуществляется по схеме триада-тетрада.

Рассмотрим отрезки числовых рядов в этих системах:

$$2: \quad 0,1,10,11,100,\dots$$

$$8: \quad 0,1,2,3,4,5,6,7,10,11,12,\dots$$

$$16: \quad 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,11,\dots$$

триада – число в восьмеричной системе, выраженное через двоичную (0 и 1).

тетрада – число в шестнадцатиричной системе, выраженное через двоичную (0 и 1).

$$\text{Пример. } 12_{10} = 14_8 = 001100_2 = C_{16}$$

Практическая схема преобразования – десятичное число  $\rightarrow$  восьмеричное, а далее замена по правилу триады. Полученное число разбиваются на тетрады.

**Алгоритм** (алгорифм) (латинская транслитерация имени математика аль-Хорезми – algorithmi) – конечный набор правил, позволяющих чисто механически решать любую конкретную задачу из некоторого класса однотипных задач. При этом подразумевается, что исходные данные могут изменяться в определенных пределах (массовость алгоритма); что процесс применения правил к исходным данным (путь решения задачи) определен вполне однозначно (детерминированность алгоритма); что на каждом шаге процесса применения правил известно, что считать результатом этого процесса (результативность алгоритма). В теории алгоритмов это интуитивно-содержательное понятие алгоритма уточняется в строгих понятиях: "рекурсивная функция", "машина Тьюринга", "нормальный алгорифм" и пр.<sup>5</sup>

---

<sup>5</sup> См.: Ершов Ю.Л., Палютин Е.А. Математическая логика. // Учебное пособие. М., 1979. 320 с.

**Программа** (греч. – programma – объявление, распоряжение). Программа вычислительной машины – описание алгоритма решения задачи, заданное на языке вычислительной машины.

Блок-схема (программы) – графическое изображение алгоритма решения задачи. Построение блок-схемы осуществляется посредством набора графических элементов:

- овал – начало (конец) программы
- прямоугольник с текстом, записанным внутри него – отдельный (законченный) шаг этап программы;
- параллелограмм с текстом внутри него – ввод-вывод данных в программе;
- ромб с текстом, записанным внутри него – логическое условие
- линии со стрелками – переход после выполнения заданного (определенного) действия.

Алгоритм программы строится таким образом, чтобы результат решения получался за конечное число шагов (вычислений), либо применением конечного числа итераций.

### **Индивидуальное задание**

1. Преобразовать целое число из 10 с/с в {2, 3, 4, 5, 6, 7, 8, 9}
2. Преобразовать правильную дробь из 10 с/с в {2, 3, 4, 5, 6, 7, 8, 9}
3. Преобразовать дробное число из 10 с/с в {2, 3, 4, 5, 6, 7, 8, 9}
4. Преобразовать целое число из 10 с/с в 16-ю с/с, используя схему триада-тетрада.
5. Нарисовать блок-схему алгоритма преобразования десятичного дробного числа в двоичное представление.

### **Структура отчета по лабораторной работе**

1. Титульный лист (см. Приложение );
2. Название лабораторной работы;
3. Цель лабораторной работы;
4. Краткие теоретические сведения, необходимые для выполнения лабораторной работы;
5. Индивидуальное задание;
6. Решение (по варианту) с пояснениями;



7. Выводы, обобщения, критический анализ и комментарии к результатам, полученным в процессе выполнения лабораторной работы;
8. Список литературы, использованной для выполнения лабораторной работы.

### **Оформление отчета по лабораторной работе**

Текст отчета пишется с одной стороны стандартного машинописного листа формата А4. Межстрочное расстояние 1 интервал, размер шрифта – 12, шрифт – TimesNewRoman. Слева и справа на полях оставляются поля: слева 3 см., справа 1 – 1.5 см., сверху и снизу 2.0 – 2.5 см. Абзацный отступ -1.25 см.

Список литературы должен оформляться согласно существующим правилам.

### **Рекомендуемая литература и источники информации**

1. Бордовский Г.А. и др. Информатика в понятиях и терминах. Кн. для учащихся ст.классов ср.школ. М., 1991. 208 с.
2. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Общая информатика. Уч.пособие для ср.школы. М., 1998. 592 с.
3. Погорелый С.Д., Слободянюк Т.Ф. Программное обеспечение микропроцессорных систем: Справочник. Киев, 1989. 301 с.
4. Абдулгалимов А.М. и др. Методические указания к выполнению лабораторных работ по дисциплине «Информационные системы». Махачкала, ДГТУ, 2005, - 35 с.
5. Абдулгалимов А.М., Денгаев А.М. Информационные системы и технологии: учебное пособие.- Махачкала, ДГТУ, 2016.-172 с.
6. Электронно-библиотечная система ООО «Издательство Лань» ([www.e.lanbook.com](http://www.e.lanbook.com)).2020 .
7. Электронно-библиотечная система (ЭБС) IPRbooks ([www.iprbooks.ru](http://www.iprbooks.ru)). 2020.
8. <http://window.edu.ru>

Приложение: титульный лист отчета по лабораторной работе

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ  
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ»**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
ПРИКЛАДНОЙ ИНФОРМАТИКИ  
В ЭКОНОМИЧЕСКИХ СИСТЕМАХ В ЭКОНОМИКЕ**

**ОТЧЕТ**  
по лабораторной работе № \_\_\_\_\_  
по дисциплине «Информационные системы и технологии»

**Выполнил (а):**

студент(ка) \_\_\_\_\_  
фамилия и инициалы студента

\_\_\_\_\_   
номер группы, курс

\_\_\_\_\_   
факультет

**Руководитель:**

\_\_\_\_\_   
ученая степень, звание, (должность)

\_\_\_\_\_   
фамилия и инициалы

**МАХАЧКАЛА – 2020 г.**

## 2. Лабораторная работа №2

### Тема: «Составление программы перевода числа из одной системы счисления в другую»

Цель: составление и реализация на ЭВМ программы перевода числа из десятичной системы счисления в двоичную.

Блок-схема программы представлена ниже на рисунке. Программирование производится на любом алгоритмическом языке, который изучался ранее, или которым владеет студент.

Требования к программе:

- программа должна быть рабочей, то есть переводить произвольно задаваемые числа в интерактивном режиме;
- быть именной – в начале работы на экране дисплея высвечиваются данные студента – фамилия И.О., № группы.

Отчет по лабораторной работе должен иметь стандартное оформление и содержать текст программы на алгоритмическом языке. Работа считается сданной, если программа правильно переводит числа.

### Блок-схема алгоритма перевода числа из одной системы счисления в другую

Абдулгалимов А.М.

#### Пояснения к блок-схеме

X-переводимое число;

q –основание системы счисления, в которую переводится число;

n –степень точности перевода дробной части числа;

Dг\_c –дробная часть числа;

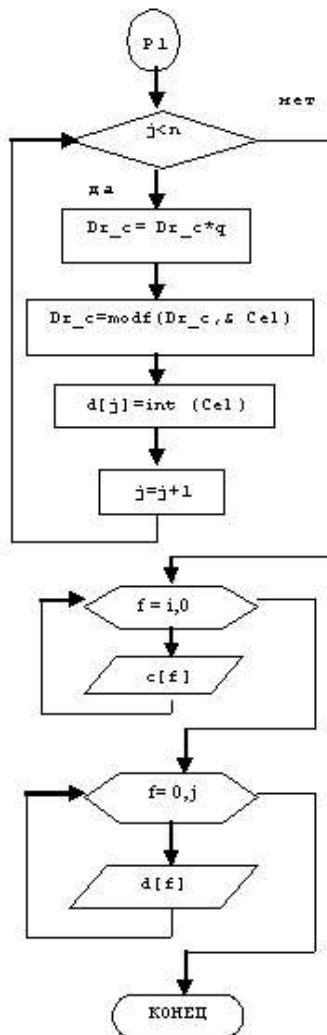
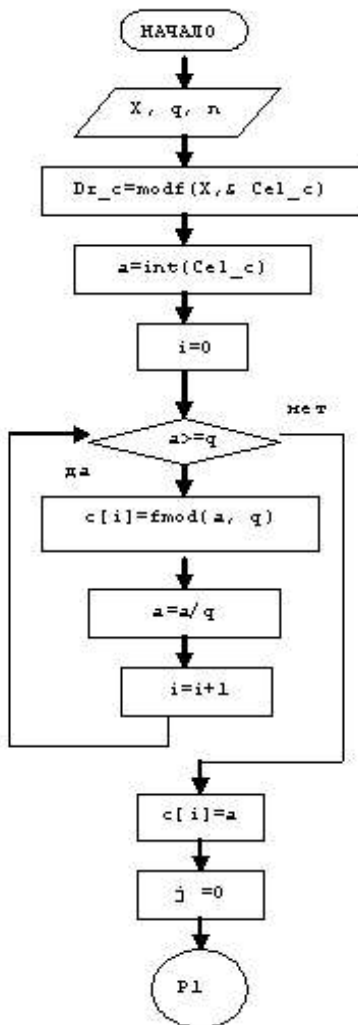
Сel\_c- целая часть числа;

i – счетчик разрядов при переводе целой части числа;

j – счетчик разрядов при переводе дробной части числа;

c(i) – массив разрядов целой части числа;

d (j) - массив разрядов дробной части числа.



**Программа перевода числа из одной  
системы счисления в другую**  
Абдулгалимов А.М.

// Abdul4.cpp: определяет точку входа для консольного приложения.  
// Программа для перевода целой и дробной частей числа из 10-й с/с в  
2-ю с/с

```
#include "stdafx.h"
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    cout << "VVEDITE chislo and tochnost\n";
    int c[10],d[10];
    int i,b,n,f,a,j;
    double cel,e,m,x;
    cin >> x >> n;
    e = modf(x, &m);
    a = int(m);
    i = 0;
    while (a >= 2)
    {
        c[i] = fmod(a, 2);
        b = a / 2;
        a = b;
        i++;
    }
    c[i] = a;
    j = 0;
    while(j<=n)
    {
        e *= 2;
        e = modf(e, &cel);
        d[j] = int(cel);
        j++;
    }
    for (f = i;f >= 0;f--)
        cout << c[f];
    cout << ", ";
}
```

```
for (i = 0; i<j;i++)  
    cout << d[i];  
return 0;  
}
```

### Индивидуальное задание

Индивидуальное задание студенту заключается в составлении и реализации на ЭВМ программы перевода смешанного числа из десятичной системы счисления в двоичную. Конкретное число студенту выдает преподаватель.

#### Рекомендуемая литература и источники информации

1. Бордовский Г.А. и др. Информатика в понятиях и терминах. Кн. для учащихся ст.классов ср.школ. М., 1991. 208 с.
2. Симонович С.В., Евсеев Г.А., Алексеев А.Г. Общая информатика. Уч.пособие для ср.школы. М., 1998. 592 с.
3. Погорелый С.Д., Слободянюк Т.Ф. Программное обеспечение микропроцессорных систем: Справочник. Киев, 1989. 301 с.
4. Электронно-библиотечная система (ЭБС) «Ай-букс.ру/ibooks.ru» ([www.ibooks.ru](http://www.ibooks.ru)).
5. Электронно-библиотечная система ООО «Издательство Лань» ([www.e.lanbook.com](http://www.e.lanbook.com)).
6. <http://window.edu.ru>

### 3. Лабораторная работа № 3

#### Тема: «Создание базы данных в MS SQL Server»

Предполагается, что на Вашем компьютере уже установлен пакет профессиональных программ «MS SQL Server 2014». Для создания базы данных в MS SQL Server 2014 проще всего воспользоваться средством управления *SQL Server Management Studio (SSMS)*. Условно говоря, SSMS - это графический интерфейс пользователя (оболочка, среда) для управления MS SQL Server.

Итак, чтобы создать новую базу идём Пуск>Программы>MS SQL Server>Среда SQL Server ManagementStudio: для создания новой базы «с нуля» в ServerManagementStudio становимся на меню «**Базы данных**», нажимаем правую кнопку и выбираем в контекстном меню «**Создать базу данных**»: появляется форма, в которой нужно указать имя вашей базы данных - пусть в нашем примере это будет учебная база **OOO\_Kaxul**: в столбце «Путь» можно указать локализацию базы данных, если хотите, чтобы она отличалась от принятой по умолчанию: нажимаем на кнопку ОК и видим в списке баз данных нашу вновь только что созданную базу.

Создать новую базу можно и программным путём, используя конструкцию **Create DataBase** на языке Transact-SQL, например:

```
USE master;
GO
CREATE DATABASE OOO_Kaxul
ON
{NAME = Sales_dat,
  FILENAME = 'C:\MyBase\OOO_KaxulDat.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5}
LOG ON
{ NAME = Sales_log,
  FILENAME = 'C:\MyBase\OOO_KaxulLog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB};
GO
```

Выполняем созданный программный запрос, и приходим к аналогичному результату, какого мы достигли «ручным» путём. Таким образом, база данных создана. Пока она пустая. Она должна содержать таблицы базы данных. Конкретно созданием таблиц займемся в лабораторной работе № 4, а здесь, не вдаваясь в подробности объявления ключей в таблицах, заполним ими созданную базу данных, т.е. мы будем создавать таблицы в этой базе данных, так как информация в базе данных хранится в таблицах, которые являются отображением некоторых логических сущностей. В нашей учебной базе **ООО\_Kaxul**, которая будет моделировать работу некой коммерческой фирмы по продаже товаров, как минимум, будут необходимы пять таблиц:

1. *Sotrudniki* - в этой таблице будет храниться информация о сотрудниках - Справочник сотрудников фирмы;
2. *Tovary* - будет содержать информацию о товарах - Справочник товаров;
3. *Zakazy* - сюда будем заносить информацию о сделанных заказах - оперативная таблица о сделанных заказах;
4. *Zakazchiki* - будет хранить информацию о заказчиках - Справочник заказчиков;
5. *Postavschiki* - хранит информацию о поставщиках товаров - Справочник поставщиков

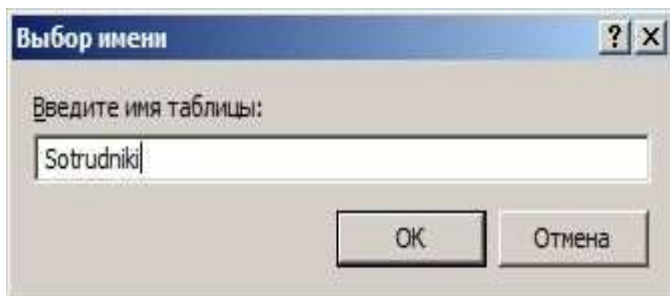
Таким образом, мы прикинули приблизительно номенклатуру таблиц нашей базы данных **ООО\_Kaxul**. Конкретную структуру таблиц с необходимыми полями будем задавать во время их создания.

Займемся теперь созданием таблиц. Опять запускаем SQL Server Management Studio, становимся на нашу БД, выбираем пункт меню «Таблицы» и по правой кнопке щёлкаем по пункту меню «Создать таблицу». Для начала попробуем создать таблицу **Sotrudniki**, в которой будем хранить данные о сотрудниках. Пусть для начала она будет иметь три поля – SotrID (идентификатор сотрудника), FIO (ФИО сотрудника) и Zarplata (зарплата). Для выполнения этой задачи в колонках «Имя столбца» и «Тип данных» заносим соответствующие реквизиты полей таблицы.

Для сохранения введённой информации о структуре таблицы нажимаем правой кнопкой по ярлыку с именем таблицы по умолчанию.

В следующей форме задаём имя таблицы:





Теперь, после обновления, в Обозревателе объектов можно увидеть в списке таблиц только что созданную таблицу **Sotrudniki**:

Нашу первую таблицу в MS SQL Server успешно создали! (пока не обращайтесь внимание на префикс **dbo** в имени таблицы. Он означает владельца таблицы - **database owner**).

Если мы что-то забыли, всегда можно вставить в структуру таблицы новый столбец. Для этого становимся на нужную таблицу, нажимаем на правую кнопку мыши и выбираем в контекстном меню «Создать столбец». Затем становимся на строке столбца, перед которым желаем вставить новый и по правой кнопке кликаем по «Вставить столбец». Далее действовать по инструкциям, высвечиваемым на экране.

### Индивидуальное задание

Создать базу данных для моделирования работы торговой фирмы (название базы данных придумается самим студентом), занимающейся продажей товаров определенного вида в соответствии с вариантом индивидуального задания (см. ниже).

В базе данных необходимо создать пять таблиц с подобными названиями, приведенными выше в примере. Конкретную структуру каждой таблицы с необходимыми полями предлагает студент во время их создания.

**Варианты индивидуального задания для студентов:**

№ варианта	1	2	3	4	5	6	7
Вид товара	Мужская одежда	Мужская обувь	Женская одежда	Женская обувь	Детская одежда	Детская обувь	Спортивная одежда

8	9	10	11	12	13	14
Рабочая спецодежда	Хлебные продукты	Хозтовары	Бытовая химия	Молочные продукты	Мясные продукты	Соки-воды

**Рекомендуемая литература и источники информации**

1. Бондарь А.Г. Microsoft SQL Server 2012. - СПб.: БХВ-Петербург, 2013. - 608 с.
2. Microsoft SQL Server 2014 (Электронная версия, 2015 г.)
3. <http://window.edu.ru>
4. Электронно-библиотечная система (ЭБС) «Ай-букс.ру/ibooks.ru» ([www.ibooks.ru](http://www.ibooks.ru)).
5. Электронно-библиотечная система ООО «Издательство Лань» ([www.e.lanbook.com](http://www.e.lanbook.com)).

#### 4. Лабораторная работа № 4

### Тема: «Создание и удаление таблицы базы данных в MS SQL Server»

#### 1. Создание и удаление таблицы диалоговыми средствами Management Studio

Запускаем Management Studio, нажав на его ярлык на рабочем столе. Далее действуем по следующей схеме: в окне **Обозреватель объектов** (Object Explorer) раскрыть **Базу данных (Database)** двумя кликами мыши и раскрыть двумя кликами мыши базу данных **i431** (эта база данных нами создана средствами языка Transact-SQL ранее), затем щелкнуть правой кнопкой мыши по папке **Таблицы (Tables)** и в контекстном меню выбрать элемент **Создать (New Table)**, затем выбрать элемент **Таблица (Table)** во всплывающем окне.

Появится окно, где можно описывать столбцы создаваемой таблицы и их характеристики (см. рис. 1):

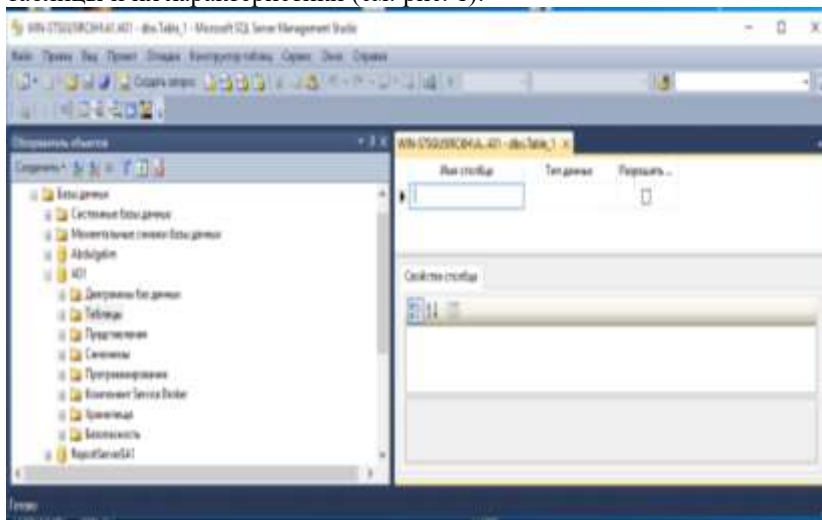


Рис. 1. Описание столбцов создаваемой таблицы

Создадим таблицу, например, Facultet со следующей структурой:

Таблица Facultet

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
NumFa	Первичный ключ	int	нет
NameFa	Название факультета	varchar(60)	да

Вводим имя первого столбца NumFa (первичный ключ – в этом столбце хранится номер записи), выбираем из выпадающего списка тип данных int. Первичный ключ не может быть пустым, поэтому и оставляем неотмеченным поле «Разрешить значения null». Затем аналогичным образом вводим имя второго столбца NameFa, задаем тип, разрешаем полю иметь значение null. Таблица принимает следующий вид:

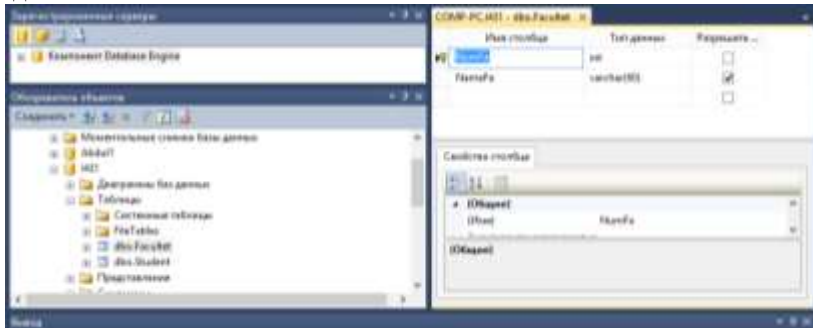


Рис. 2. Структура таблицы Facultet

Теперь необходимо указать, что поле NumFa будет являться первичным ключом. Правой кнопкой мыши щелкаем по этому полю и выбираем «Задать первичный ключ»:

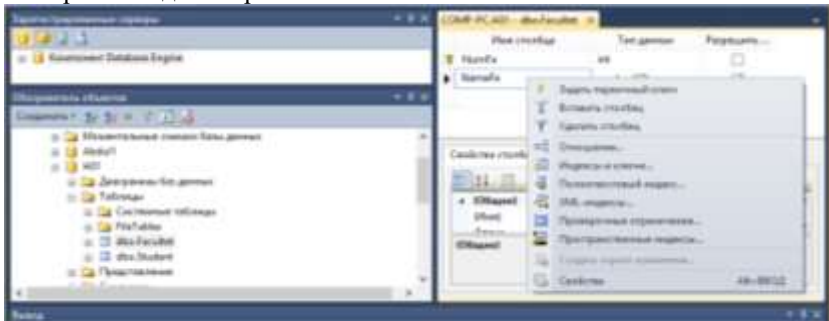


Рис. 3. Задание первичного ключа в таблице Facultet.

Теперь нужно сохранить созданную таблицу. Щелкните правой кнопкой мыши по заголовку вкладки окна создания структуры таблицы и в контекстном меню выберите элемент **Сохранить Table\_1**. Появится окно задания имени таблицы, где нужно ввести имя таблицы **Facultet** и щелкнуть мышью по кнопке **ОК**. Таблица будет сохранена с этим именем в базе данных (в данном случае в текущей базе данных **i431**). После этого таблица должна появиться в обозревателе объектов:

**dbo.Facultet.** Теперь можно перейти к заполнению этой таблицы. Для этого нужно в обозревателе объектов выбрать эту таблицу и в контекстном меню нажать **«Изменить первые 200 строк»**. В результате частичного заполнения таблицы записями она будет выглядеть следующим образом: см. рис. 4.

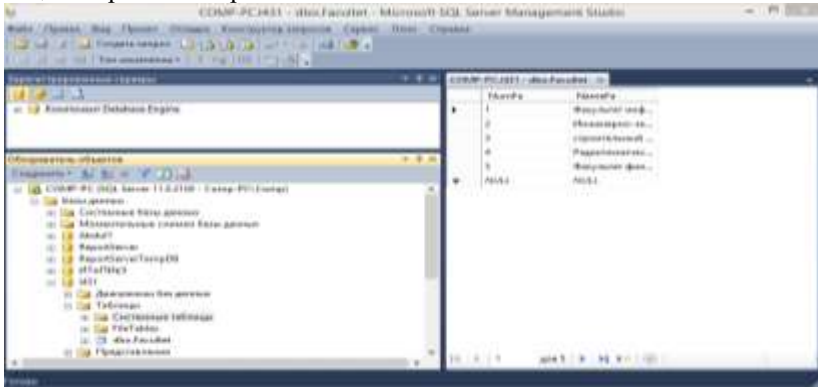


Рис. 4. Заполненная таблица Facultet.

Закрываем эту таблицу: щелкните правой кнопкой мыши по заголовку вкладки окна заполнения таблицы и в контекстном меню выберите элемент **Заккрыть**.

Далее аналогичным образом создадим таблицу Student со следующей структурой:

Таблица Student

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
NumSt	Первичный ключ	int	нет
NameSt	Фамилия, имя и отчество студента	varchar(60)	да
NumFa	NumFa (внешний ключ, ссылается на первичный ключ таблицы Facultet)	int	нет

*Внешний* ключ используется в комбинации с первичным для связывания двух таблиц по общему столбцу (столбцам). К примеру, можно связать таблицы Facultet и Student по столбцу NumFa, который присутствует в обеих таблицах. Поскольку поле NumFa таблицы Facultet является его первичным ключом можно использовать поле NumFa таблицы Student в качестве внешнего ключа, который свяжет эти две таблицы. После организации такого ограничения бу-

дет невозможно добавить запись в таблицу Student, если в таблице Facultet нет записи с соответствующим значением NumFa. Кроме того, при отсутствии каскадирования (специальные условия, накладываемые на внешний ключ при его создании) невозможно удалить запись из таблицы Facultet при наличии связанных с ней записей в таблице Student, поскольку нельзя оставлять факультет без информации о студенте. Для создания описанного ограничения внешнего ключа в Management Studio выполните следующие шаги:

1. В контекстном меню папки «Ключи» таблицы Student выберите команду «Создать внешний ключ...» (см. рис. 5):

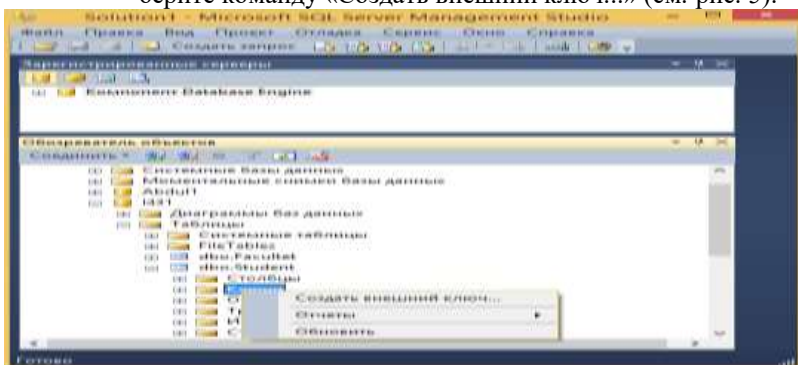


Рис.5. Создание внешнего ключа в таблице Student.

2. В открывшемся окне «Связи по внешнему ключу» заполните следующие поля:
  - ✓ (Имя): FK\_Student\_Student
  - ✓ *Спецификация таблиц и столбцов*: для заполнения данного блока щелкните на кнопке с многоточием и в появившемся окне «Таблицы и столбцы» в качестве таблицы первичного ключа выберите Facultet, а полей связи – NumFa (см. рис. 6):

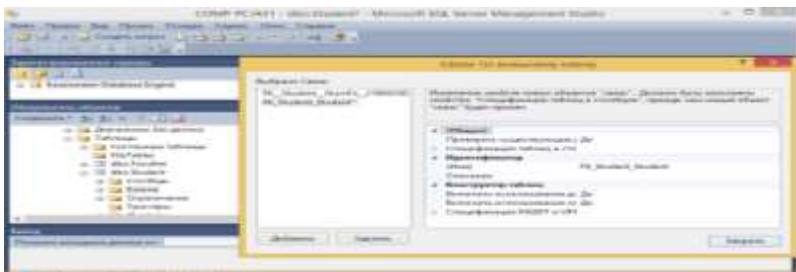


Рис. 6. Связи по внешнему ключу в таблице Student.

3. Закройте все открывшиеся окна с сохранением изменений.

Таблица **Student** также заполняется данными в соответствии с соблюдением правил заполнения таблиц с первичными и внешними ключами. Вид заполненной таблицы представлен на рис. 7:

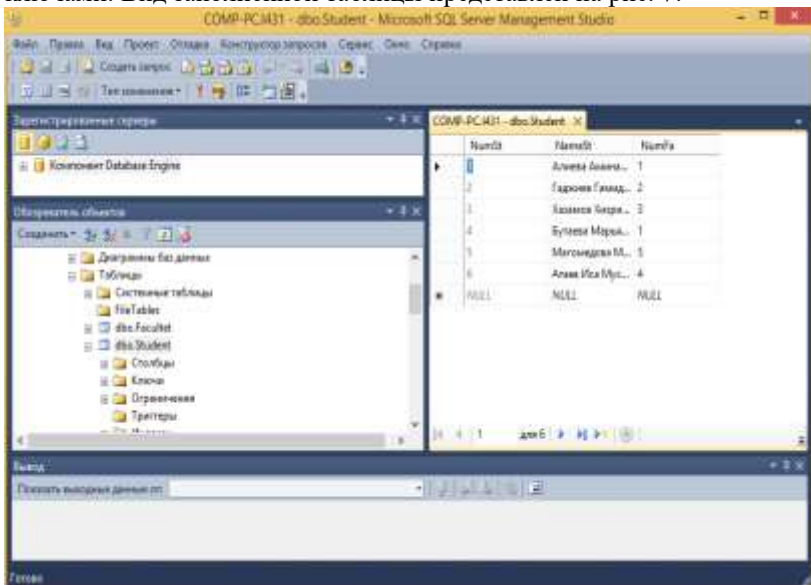


Рис.7. Заполненная таблица Student.

Удаление существующей в базе данных таблицы можно осуществить, как сказано ниже, с помощью оператора языка Transact-SQL **DROP TABLE** или же при использовании диалоговых средств в программе Management Studio: **Обозреватель объектов**→**Базы данных**→**I431** (пример базы данных)→**Таблицы**→**dbo.Student** (пример таблицы). Кликнуть на этой таблице мышью и в появившейся заставке выбрать команду «Удалить».

## 2. Операторы создания и удаления таблицы в языке Transact-SQL

В одной базе данных может присутствовать более двух миллиардов различных объектов, включая таблицы.

Для создания таблицы в языке Transact-SQL используется оператор **CREATE TABLE**, синтаксис которого в простейшем случае представлен ниже:

```

CREATE TABLE <имя таблицы>
(
  <имя столбца 1> <тип данных> [NOT NULL] [DEFAULT <значение
по умолчанию>],
  <имя столбца 2> <тип данных> [NOT NULL] [DEFAULT <значение
по умолчанию>],
  ...
);

```

Проиллюстрируем это на примере создания двух таблиц: «Факультет» (Facultet), «Студент» (Student), имеющих соответственно следующие структуры:

Таблица Facultet

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
NumFa	Первичный ключ	int	нет
NameFa	Название факультета	varchar(60)	да

В реляционных базах данных первичный ключ используется как уникальный идентификатор записи. Это поле является обязательным, оно используется для связи таблиц по внешним ключам. Первичный ключ должен иметь целочисленный тип (в данном случае - *int*). Во втором поле будет храниться название города - некоторая строка, поэтому мы выбираем для этого поля тип *varchar(30)*. Число в скобках означает максимальное число символов в строке.

Таблица Student

Имя поля (столбца)	Содержание	Тип данных	Возможность содержать NULL
NumSt	Первичный ключ	int	нет
NameSt	Фамилия, имя и отчество студента	varchar(60)	да
NumFa	NumFacultet (внешний ключ, ссылается на первичный ключ таблицы Facultet)	int	нет



Эта таблица содержит поле NumFa которое ссылается на первичный ключ таблицы Facultet.

Начнем с создания таблицы Facultet, используя приведенный выше синтаксис команды создания таблицы. Введем запрос:

```

/* создание таблицы Факультет */
USE i431 -- определяем базу данных, в которую
          входит таблица
CREATE TABLE Facultet
(
  NumFa INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- первич-
  ный ключ
  NameFa VARCHAR(60) -- название факультета
)

```

В обозревателе объектов видим, что таблица действительно создана. Слово IDENTITY(1,1) добавлено, чтобы поле первичного ключа NumFacultet автоматически нумеровалось начиная с единицы (фактически, эта конструкция определяет спецификацию идентифицирующего столбца).

Таким же образом можно создать и таблицу Student. Эта таблица содержит поле NumFa, которое ссылается на первичный ключ таблицы Facultet. Чтобы создать такую таблицу, необходимо выполнить запрос:

```

/* создание таблицы Студент */
USE i431 -- определяем базу данных, в которую
          входит таблица
CREATE TABLE Student
(
  NumSt INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- первич-
  ный ключ
  NameSt VARCHAR(60), -- фамилия, имя и отчество
  студента

  NumFa INT FOREIGN KEY REFERENCES Facultet (NumFa) -- фа-
  культет,
  -- ссылка по внешнему ключу на поле NumFa таблицы Facultet
)

```

**Примечание.** Ссылку можно создать только на существующую таблицу.

Синтаксис оператора удаления таблицы в упрощенном виде выглядит так: **DROP TABLE** <имя таблицы>; . В одном операторе можно удалить несколько таблиц, отделяя их имена запятыми.

### Порядок выполнения работы

1. Ознакомиться со своим вариантом задания.
2. Разработать схемы таблиц для предметной области из индивидуального задания. Таблиц должно быть не менее трех, например, две таблицы сущностей с первичными ключами и связующая (основная) таблица, включающая внешние ключи к первым двум таблицам. Согласовать с преподавателем схемы таблиц.
3. Запустить пакет профессиональных программ MS SQL Server 2014.
4. Создать базу данных.
5. Создать таблицы.
6. Заполнить таблицы значениями. В каждой таблице должно быть не менее десяти записей.

### Варианты индивидуальных заданий для студентов

Ниже перечислены варианты индивидуальных заданий для лабораторных работ. Каждое задание содержит неструктурированный перечень терминов из некоторой предметной области, который составляет основу для проектирования базы данных, но не является исчерпывающим. Поэтому в зависимости от задания студент должен отобрать необходимые понятия из предметной области, а при необходимости добавить новые, исходя из собственных представлений о данной предметной области, и самостоятельно выявить сущности, атрибуты и ассоциации, которые должны войти в базу данных. Предлагаются следующие варианты:

1. Химическое производство; место хранения, наименование изделия, план выпуска изделий, состав изделия, сырье, свойства сырья, количество в наличии, ФИО ответственного за хранение вещества.
2. Банк; наименование и адрес отделения банка, номер сейфа, номер ключа от сейфа, местонахождение ключа (банк, клиент), ФИО клиента, доверенность на использование сейфа, доверенное лицо, дата посещения сейфа.
3. Банк; наименование и адрес отделения банка, ФИО клиента, адрес клиента, документы, удостоверяющие личность, номер счета, ба-

- ланс счета, валюта счета; код финансовой операции, сумма операции, дата операции.
4. Банк; наименование и адрес отделения банка, номер и адрес банкомата, количество денег в банкомате, вид валюты; время инкассации банкомата, время подкрепления банкомата; ФИО инкассатора.
  5. Почта; наименование и адрес отделения почты, номер посылки, время прихода посылки в отделение, время пересылки в другое отделение, ФИО получателя, адрес получателя, дата получения посылки, ФИО отправителя, адрес отправителя.
  6. Производство; цех, ФИО работника, номер бригады, состав бригады, бригадир, наименование смены, время начала и завершения смены, наименование изделия, количество изделий выпущенных в смену, количество брака за смену.
  7. Производство; наименование изделия, комплектующее изделие, номер заказа, наименование заказчика, количество произведенных изделий по заказу.
  8. Учет рабочего времени; ФИО сотрудника, должность, название проекта, задача по проекту, время выполнения задачи, отчет сотрудника за неделю.
  9. Библиотека; автор книги, название, год издания, цена, аннотация; номер читательского билета, ФИО, адрес и телефон читателя, дата выдачи книги читателю и дата сдачи книги читателем.
  10. Отдел кадров университета; ФИО, адрес и должность преподавателя; название дисциплины, количество часов; название кафедры, ФИО заведующего кафедрой, название факультета, ФИО декана, группа, специальность, количество студентов в группе.
  11. Склад; код товара, название товара, количество на складе, стоимость единицы товара, описание товара; номер и ФИО поставщика товара, срок поставки и количество товаров в поставке.
  12. Производство; код изделия, название изделия, назначение изделия; код, название, адрес и телефон предприятий, выпускающих изделия; год выпуска и объем выпуска данного изделия предприятием.
  13. Сеть магазинов; ФИО директора магазина, адрес, телефон магазина; ФИО, адрес, телефон поставщика, стоимость поставки данного поставщика в данный магазин.
  14. Сеть автомастерских; номер водительских прав, ФИО, адрес и телефон владельца автомобиля; ФИО, адрес, телефон и квалификация механика; номер, марка и цвет автомобиля; название, адрес и телефон ремонтной мастерской.

15. Деканат; наименование специальности, код группы, ФИО, дата рождения, домашний адрес, телефон студента; название, количество часов и вид контроля предметов, оценки каждого слушателя по каждому предмету в каждую сессию.
16. Договорная деятельность организации; шифр договора, наименование организации, сроки выполнения, сумма договора, примечания, вид договора; ФИО, адрес, телефон, должность, оклад сотрудников, сроки работы данного сотрудника по данному договору.
17. Регистратура; фамилия, имя, отчество, дата рождения пациента; ФИО, должность и специализация лечащего врача, диагноз, номер больничного листа, срок потери трудоспособности.
18. Телефонная служба; номер телефонной станции, адрес телефонной станции, номер абонента, фамилия абонента, адрес, дата установки, тариф оплаты, фактическая оплата.
19. Реестр предприятий; наименование предприятия, адрес, дата регистрации, вид собственности, количество работников, вид деятельности, ФИО директора, обслуживающий банк, адрес банка, финансовый баланс предприятия.
20. Транспортное предприятие; вид транспорта, номер маршрута, перечень остановок по маршруту, время передвижения между остановками, ФИО водителя, марка транспортного средства, номер государственной регистрации.
21. Министерство иностранных дел; название страны, столица, площадь территории, количество населения, преобладающая национальность, ФИО посла, соседние страны, военный блок, штаб-квартира военного блока.
22. Жилищное хозяйство; номер ЖЭУ, обслуживаемые дома, номер квартиры, площадь квартиры, количество комнат, ФИО квартиросъемщика, ФИО членов семьи.
23. Аэропорт; номер рейса, пункт назначения, дата рейса, тип самолета, время вылета, время в пути, сведения о пассажире.
24. Школа; ФИО учителя, директор, заместитель директора, учитель, преподаваемая дисциплина, учащийся, родители учащегося, номер класса.
25. Спортивный комитет; спортивный клуб, ФИО спортсмена, дата рождения, страна, квалификация, соревнования, место проведения, дата проведения, команда на соревнованиях, занятое место, ФИО тренера.
26. Зоопарк; место нахождения (страна, город) зоопарка, вид животного, род, семейство, количество, корм, нормы кормления, ответ-

- ственный сотрудник, ФИО сотрудника, должность.
27. Дорожное хозяйство; название дороги, населенный пункт, расстояние между населенными пунктами, категория дороги, вид покрытия.
  28. Больница; отделение больницы, заведующий отделением, лаборатория, заведующий лабораторией, врач, специальность врача, пациент, болезнь, палата, больничный корпус.
  29. Дошкольное образование; детское образовательное учреждение, директор, воспитатель, группа, ребенок, родитель, место работы родителя, должность, адрес жительства, домашний телефон,
  30. Гараж: марка автомобиля, номер государственной регистрации, год выпуска, водитель, возраст, стаж работы, оклад, маршрут, план выезда, фактическое исполнение, заказчик.

### Рекомендуемая литература и источники информации

1. Бондарь А. Г. Microsoft SQL Server 2012. - СПб.: БХВ-Петербург, 2013. - 608 с.
2. Microsoft SQL Server 2014 (Электронная версия, 2020 г.)
3. <http://window.edu.ru>
4. Базы данных : метод. указания к лаборатор. работам / Владим. гос. ун-т имени Александра Григорьевича и Николая Григорьевича Столетовых ; сост. И. Р. Дубов. – Владимир : Изд-во ВлГУ, 2012. – 24 с.
5. Электронно-библиотечная система (ЭБС) «Айбукс.ру/ibooks.ru» ([www.ibooks.ru](http://www.ibooks.ru)).
6. Электронно-библиотечная система ООО «Издательство Лань» ([www.e.lanbook.com](http://www.e.lanbook.com)).

## 5. Лабораторная работа № 5

**Тема: «Добавление, изменение, выборка и удаление данных в таблице базы данных в MS SQL Server»**

Для добавления в таблицу одной или нескольких строк используется оператор **INSERT**. Его упрощенный синтаксис представлен ниже:

```
INSERT INTO <имя таблицы> VALUES (<значение>, ..., (<значение>);
```

При такой записи указанные в скобках после ключевого слова **VALUES** значения вводятся в поля добавленной в таблицу новой строки в том порядке, в котором соответствующие столбцы указаны при создании таблицы, то есть в операторе **CREATE TABLE**.

Например, ввод новой строки в таблицу **Student** может быть осуществлен следующим образом:

```
INSERT INTO Student VALUES (7, 'Мурадов Магомед Велиевич', 2);
```

Команда **UPDATE** позволяет **изменять**, то есть обновлять значения некоторых или всех полей в существующей строке или строках таблицы. Например, чтобы для всех университетов, сведения о которых находятся в таблице **University**, изменить рейтинг на значение 200, можно использовать конструкцию:

```
UPDATE University
SET rating = 200;
```

Для указания конкретных строк таблицы, значения полей которых должны быть изменены, в команде **UPDATE** можно использовать запрос:

```
UPDATE University
SET rating = 200
WHERE city = 'Махачкала';
```

В результате выполнения этого запроса будет изменен рейтинг только у университетов, расположенных в Махачкале.

Команда **UPDATE** позволяет изменять не только один, но и множество столбцов. Для указания конкретных столбцов, значения которых должны быть модифицированы, также используется предложение **SET**.

Оператор **SELECT** (выбрать) языка **SQL** является самым важным и самым часто используемым оператором. Он предназначен для **выборки** информации из таблиц базы данных. Упрощенный синтаксис оператора **SELECT** выглядит следующим образом:

**SELECT [DISTINCT] <список атрибутов> FROM <список таблиц>**  
**[WHERE <условие выборки>];**

В квадратных скобках указаны элементы, которые могут отсутствовать в запросе.

Ключевое слово **SELECT** сообщает базе данных, что данное предложение является запросом *на извлечение* информации. После слова **SELECT** через запятую перечисляются *наименования полей* (список атрибутов), содержимое которых запрашивается.

Обязательным ключевым словом в предложении-запросе **SELECT** является слово **FROM** (из). За ключевым словом **FROM** указывается список разделенных запятыми имен таблиц, из которых извлекается информация.

Например,

```
SELECT
NumSt, NameSt
FROM Student;
```

Любой SQL-запрос должен заканчиваться символом «;» (*точка с запятой*).

Приведенный запрос осуществляет выборку всех значений NumSt и NameSt из таблицы Student.

Если необходимо вывести значения *всех* столбцов таблицы, то можно вместо перечисления их имен использовать символ звездочка).

```
SELECT * FROM Student;
```

В данном случае результатом выполнения запроса будет вся таблица Student.

Для исключения из результата **SELECT**-запроса повторяющихся значений полей в записях используется ключевое слово **DISTINCT** (**отличный**). Если запрос **SELECT** извлекает множество полей, то **DISTINCT** *исключает* дубликаты строк, в которых значения *всех* выбранных полей идентичны.

Использование в операторе **SELECT** предложения, определяемого ключевым словом **WHERE** (где), позволяет задавать выражение условия (предикат), принимающее значение *истина или ложь* для значений полей строк таблиц, к которым обращается оператор **SELECT**. Предложение **WHERE** определяет, *какие строки указанных таблиц должны быть выбраны*. В таблицу, являющуюся результатом запроса, включаются только те строки, для которых условие (предикат), указанное в предложении принимает значение *истина*.

**Пример.**

Написать запрос, выполняющий выборку ФИО всех студентов с номером факультета 1, сведения о которых хранятся в таблице Stu-

dent.

```
SELECT NumSt, NameSt
FROM Student
WHERE NumFa=1;
```

**Удаление** строк из таблицы осуществляется с помощью команды **DELETE**.

Следующее выражение удаляет все строки таблицы Student:

```
DELETE FROM Student;
```

В результате таблица становится пустой (после этого она может быть удалена командой **DROP TABLE**).

Для удаления из таблицы сразу нескольких строк, удовлетворяющих некоторому условию, можно воспользоваться предложением **WHERE**, например:

```
DELETE FROM Student WHERE NumSt = 3;
```

Можно удалить группу строк:

```
DELETE FROM Student WHERE NUMFA = 2;
```

Синтаксис приведенных выше операторов довольно сложный. Об этом вы будете информированы в дисциплине «Базы данных».

### Порядок выполнения работы

Индивидуальное задание по лабораторной работе №5 заключается в написании запросов, используя команды **INSERT**, **UPDATE**, **SELECT**, **DELETE** по аналогии с тем, что приведено выше в теоретической части лабораторной работы, к созданным отношениям (таблицам) в лабораторной работе № 4.

### Варианты индивидуальных заданий для студентов

Варианты индивидуальных заданий для выполнения лабораторной работы № 5 соответствуют тем, которые студенты получили во время выполнения лабораторной работы № 4.

### Рекомендуемая литература и источники информации

1. Абдулгалимов А.М., Оруджев М.И. Информационная безопасность: учебное пособие с грифом УМО.- Махачкала, ДГТУ, 2011. - 167 с.
2. Абдулгалимов А.М. и др. Методические указания к выполне-



- нию лабораторных работ по дисциплине «Информационные системы». Махачкала, ДГТУ, 2005, - 35 с.
3. Трофимов В.В. Информационные технологии в экономике и управлении: учеб.- М.: Юрайт, 2014.
  4. Гаврилов М.В. Информатика и информационные технологии: учебник.- М.: Юрайт, 2014.
  5. Абдулгалимов А.М., Денгаев А.М. Информационные системы и технологии: учебное пособие.- Махачкала, ДГТУ, 2016.
  6. Косиненко Н.С., Фризен И.Г. Информационные системы и технологии в экономике [Электронный ресурс]: учебное пособие.-Режим доступа: <http://www.iprbookshop.ru/57134.html>.— ЭБС «IPRbooks».- М.: Дашков и К, Ай Пи Эр Медиа, 2017.
  7. Балдин К.В., Уткин В.Б. Информационные системы в экономике [Электронный ресурс]: учебник. Режим доступа: <http://www.iprbookshop.ru/52298.html>.— ЭБС «IPRbooks».- М.: Дашков и К, 2015.
  8. Абдулгалимов А.М. Методические указания к выполнению лабораторных работ направления подготовки бакалавров 230700.62 – «Прикладная информатика» по дисциплине «Информационные системы и технологии». – «Фактографические информационные системы и технологии». Часть 1.- Махачкала, ДГТУ, 2013.
  9. Абдулгалимов А.М., Денгаев А.М. Методические указания к выполнению лабораторных работ по дисциплине «Информационные системы и технологии» для студентов направления подготовки бакалавров 230700.62 – «Прикладная информатика». Часть 2.- Махачкала, ДГТУ, 2015
  10. Бондарь А.Г. Microsoft SQL Server 2014.- СПб.: БХВ – Петербург, 2015.
  11. Абдулгалимов А.М., Денгаев А.М. Информационные системы и технологии: Учеб. пособие/Под ред. А.М.Абдулгалимова.- Махачкала, ДГТУ, 2016.- 172 с.
  12. Бьярне Страуструп (Bjarne Stroustrup) Программирование. Принципы и практика с использованием С++. – М.: Изд-во Вильямс, издание: 2-е, 2016.- 1328 с.
  13. Голощاپов А.Л. Microsoft Visual Studio 2010; БХВ-Петербург - Москва, 2011. - 544 с.
  14. Microsoft SQL Server 2016. Электронная версия, 2017 г.
  15. Тагиев М.Х., Тагиев Р.Х. Современные мировые информационные ресурсы: учебное пособие.- Махачкала, ДГТУ, 2010.

16. Ирзаев Г.Х. Информационные системы предметных областей экономики: учебное пособие.- Махачкала, ДГТУ, 2010.- 249 с.
17. Павловская Т.А. С/С++. Программирование на языке высокого уровня.- СПб.: Питер, 2013
18. Ромель А.П., Финкова М.А., Матвеев М.Д. Windows 10. Все об использовании и настройках. Самоучитель [Электронный ресурс]. Режим доступа: <http://www.iprbookshop.ru/60646.html>.— ЭБС «IPRbooks».- СПб.: Наука и Техника, 2016
19. Электронно-библиотечная система ООО «Издательство Лань» ([www.e.lanbook.com](http://www.e.lanbook.com)). 2020
20. Электронно-библиотечная система (ЭБС) IPRbooks ([www.iprbooks.ru](http://www.iprbooks.ru)). 2020
21. <http://window.edu.ru> – единое окно доступа к образовательным ресурсам.
22. <http://www.intuit.ru> – интернет-университет.

## Оглавление

		Стр.
	Введение.....	3
1	Лабораторная работа №1. Тема: «Системы счисления»...	4
	Индивидуальное задание.....	8
	Структура отчета по лабораторной работе.....	8
	Оформление отчета по лабораторной работе.....	9
	Рекомендуемая литература и источники информации ...	9
	Приложение: титульный лист отчета по лабораторной работе.....	10
2	Лабораторная работа №2. Тема: «Составление программы перевода числа из одной системы счисления в другую».....	11
	Блок-схема алгоритма перевода числа из одной системы счисления в другую.....	11
	Программа перевода числа из одной системы счисления в другую.....	13
	Индивидуальное задание.....	14
	Рекомендуемая литература и источники информации ...	14
3	Лабораторная работа № 3. Тема: «Создание базы данных в MS SQL Server».....	15
	Индивидуальное задание.....	17
	Рекомендуемая литература и источники информации.....	18
4	Лабораторная работа № 4. Тема: «Создание и удаление таблицы базы данных в MS SQL Server».....	19
	Создание и удаление таблицы диалоговыми средствами Management Studio.....	19
	Операторы создания и удаления таблицы в языке Transact-SQL.....	23
	Порядок выполнения работы.....	26
	Варианты индивидуальных заданий для студентов.....	26
	Рекомендуемая литература и источники информации ....	29
5	Лабораторная работа № 5. Тема: «Добавление, изменение, выборка и удаление данных в таблице базы данных в MS SQL Server».....	30
	Порядок выполнения работы.....	32
	Варианты индивидуальных заданий для студентов.....	32
	Рекомендуемая литература и источники информации...	32

**Абдулгалимов Абдулгалим Минхаджевич**

**УЧЕБНО-МЕТОДИЧЕСКИЕ  
УКАЗАНИЯ**

**к выполнению лабораторных работ по дисциплине «Информационные системы и технологии» для студентов направления подготовки бакалавров 09.03.03–Прикладная информатика, профиль «Прикладная информатика в экономике». Часть 1.**