**Лабораторная работа № 1. Знакомство с инструментами СУБД Microsoft SQL Server 2008**

Создавать базы данных можно с использованием инспектора объектов Object Explorer утилиты SQL Server Management Studio, входящей в состав программного обеспечения СУБД Microsoft SQL Server 2008 (MSS), или с помощью языка Transact-SQL. В данной лабораторной работе рассматривается первый вариант.

Я создала базу данных с помощью утилиты SQL Server Management Studio, под названием ПРОДАЖИ.

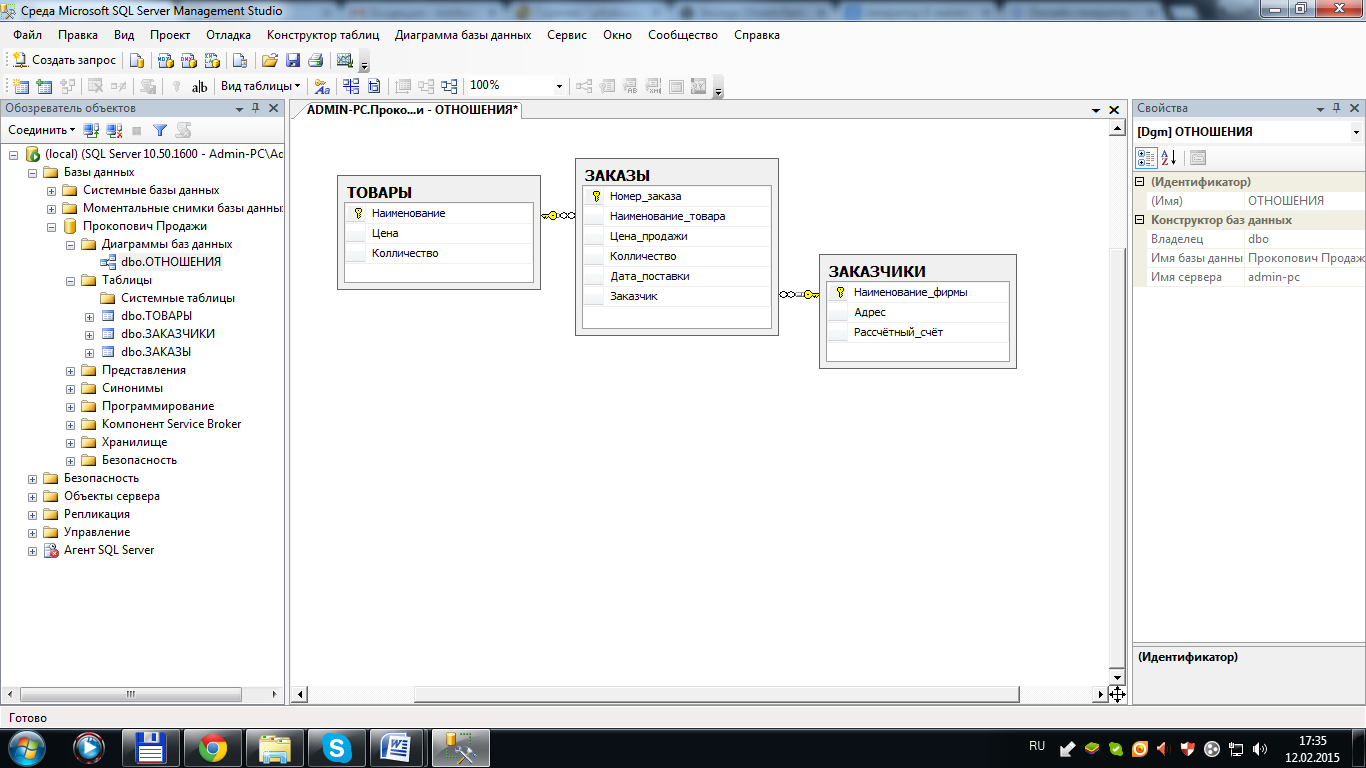
В ней содержится три таблицы.

Первая с именем ТОВАРЫ, содержит поля: Наименование (тип nvarchar(20)), Цена (тип real), Количество (тип int), первый столбец ключевой.

Вторая таблица с именем ЗАКАЗЧИКИ и полями: Наименование*\_*фирмы(тип nvarchar(20)), Адрес (тип nvarchar(50)), Расчетный\_счет (тип nvarchar(15)) , первый столбец ключевой.

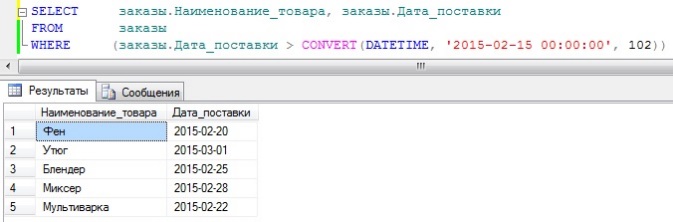
Третья таблица с именем ЗАКАЗЫ, поля: Номер\_заказа (тип nvarchar(10)), Наименование\_товара (тип nvarchar(20)), Цена\_продажи (тип real), Количество (тип int), Дата\_поставки (тип date), Заказчик (тип nvarchar(20)), первый столбец ключевой.

При объединении получается такая структура:

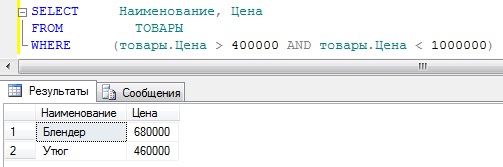


Далее нужно было создать запросы:

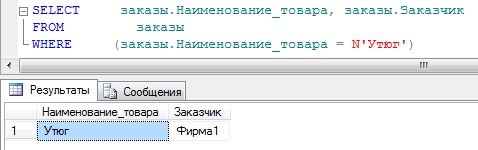
-определить товары, поставки которых должны осуществиться после некоторой даты:



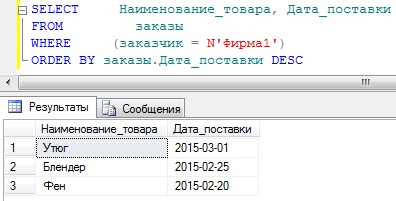
-найти товары, цена которых находится в некоторых пределах:



-определить названия фирм, заказавших конкретный товар:



-найти заказы определенной фирмы по ее названию, отсортировать их по датам поставки:

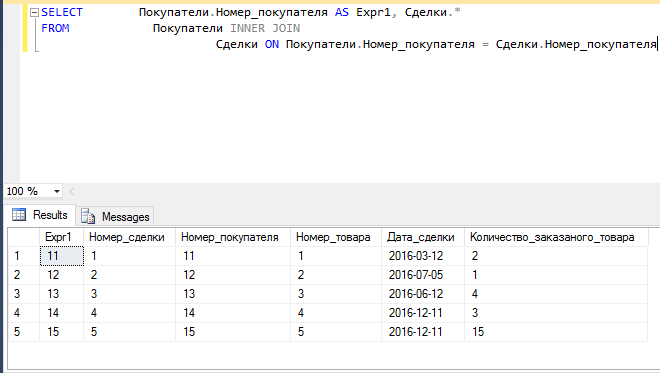
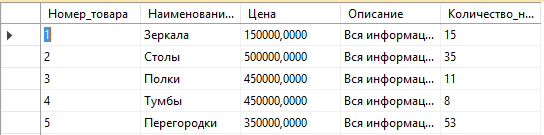
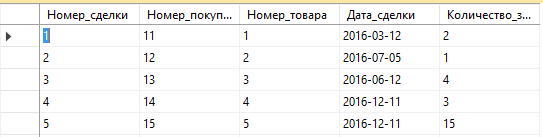
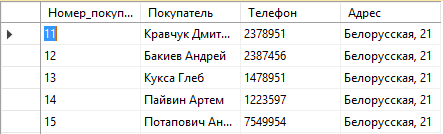
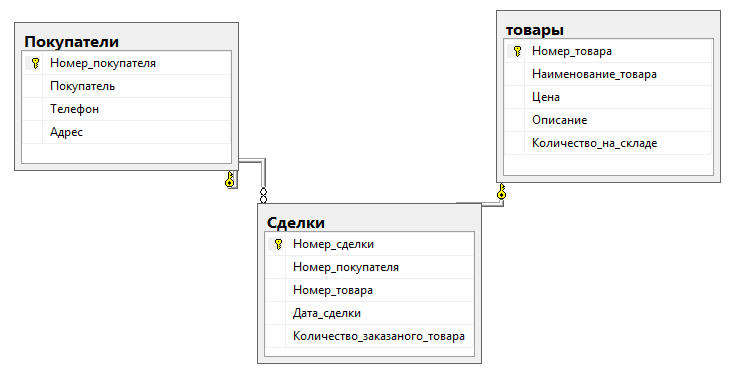


**Лабораторная работа №2. Проектирование баз данных. Нормализация**

1. В соответствии со своим вариантом, номер которого определяет преподаватель, преобразовать исходную информацию из таблицы, представленной ниже, к нормальной форме, создав три таблицы. При необходимости использовать дополнительные поля.

**Реализация изделий из стекла.** Информационные поля:

1. Номер покупателя, Покупатель, Телефон, Адрес.
2. Номер сделки, Номер покупателя, Номер товара, Дата сделки, Количество заказанного товара.
3. Номер товара, Наименование, Цена, Описание, Количество на складе.
4. Создать базу данных по нормализованным данным (имя − **X\_MyBase**,где **X** − первые буквы своей фамилии) с помощью утилиты **SSMS**, установить связи и заполнить таблицы информацией.



**Лабораторная работа №3. T-SQL - язык реляционной базы данных**

1. Создать с помощью языка **SQL** базу данных с именем, которое состоит из фамилии студента и слова **BSTU**.
2. Создать скрипт для формирования таблицы **STUDENT**, содержащей три столбца (номер зачетки, фамилию студента, номер группы) с помощью оператора **CREATE**. Просмотреть структуру таблицы.
3. Внести изменения в структуру таблицы с помощью оператора **ALTER**: добавить столбец **Дата\_поступления** (с помощью ключевого слова **Add**), посмотреть новую структуру и удалить столбец(ключевое слово **Drop**).
4. С помощью оператора **INSERT** ввести семь строк в таблицу **STUDENT**. Строки должны содержать разные фамилии студентов.
5. Использовать оператор SELECT для выбора всех строк и столбцов таблицы STUDENT. Проанализировать результат.

Написать оператор SELECT, выбирающий все строки одного столбца NAME.

Написать оператор SELECT, выбирающий все строки двух столбцов (NZach, NAME) таблицы.

Посчитать количество строк в таблице.

1. С помощью SQL Server Management Studio удалить таблицу STUDENT и создать заново (как в лабораторной работе № 1). Заполнить информацией 10 строк.

С помощью оператора UPDATE изменить значение столбца GRUP на 5 во всех записях таблицы. Удалить строку, в которой значение столбца NZach равно одному из введенных чисел. Проверить результат, используя Select.

Удалить таблицу STUDENT с помощью оператора Drop.

1. С помощью оператора CREATE создать таблицу STUDENT, в которой столбец NZach первичный ключ, столбец NAME не может принимать значение NULL.

Продемонстрировать действие ограничений, введенных для таблицы STUDENT, с помощью операторов INSERT и UPDATE.

Удалить таблицу STUDENT. Создать ее снова, но добавить новый столбец с именем POL (пол) с ограничением CHECK. Продемонстрировать действие ограничений с помощью операторов INSERT и UPDATE. Удалить таблицу STUDENT.

1. Запустить утилиту SQLCMD с параметром, позволяющим вывести в окно консоли краткую инструкцию о применении утилиты. Ознакомиться с инструкцией.

use MikhnyukBSTU

CREATE TABLE STUDENT

(

Номер\_зачетки int

constraint PK\_STUDENT primary key,

Фамилия\_студента varchar(50),

Номер\_группы int not NULL

);

ALTER Table STUDENT ADD Дата\_поступления date; --Добавляем столбец в таблицу

alter table STUDENT ADD Пол char(1) default 'м' check/\*ограничение\*/ (Пол in('м', 'ж'));

INSERT into STUDENT (Номер\_зачетки, Фамилия\_студента, Номер\_группы) --Вводим данные

Values (753335557, 'Сакович', 2),

(74312352, 'Пайвин',2),

(79852258, 'Куревич', 1),

(15972269, 'Касатый',2),

(59842345, 'Костюченко', 2),

(87337473, 'Вабищевич',1),

(69111738, 'Цедрик',1),

(86476392, 'Николаенко', 2),

(82246873, 'Степуро',2),

(11146462, 'Бельмач', 2);

SELECT \* From STUDENT; --поиск информации

SELECT Номер\_зачетки, Номер\_группы From STUDENT;

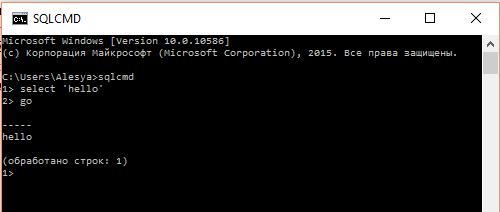
SELECT COUNT (\*) From STUDENT; -- подсчитать количество строк

UPDATE STUDENT set Номер\_группы=Номер\_группы+5; --изменяем строки

DELETE From STUDENT WHERE Номер\_группы like '6'; --удаляем строки

Select Фамилия\_студента from STUDENT;

Запустить утилиту **SQLCMD** с параметром, позволяющим вывести в окно консоли краткую инструкцию о применении утилиты. Ознакомиться с инструкцией



use master;

create database lab3;

use lab3;

create table Tovar(

Наименование\_товара nvarchar(50) primary key not null, Цена money, Описание nvarchar(50), Кол\_на\_складе int);

insert into Tovar(Наименование\_товара,Цена,Описание,Кол\_на\_складе)

values ('Зеркало',5400,'Цветное',50), ('Хрусталь',2000,'Разноцветный', 100), ('Столешница', 8000,'кругаля',20);

create table Zakazchiki(

Покупатель nvarchar(50) primary key not null, Телефон nvarchar(50), Адрес nvarchar(50));

insert into Zakazchiki(Покупатель,Телефон,Адрес)

values ('Слепцова Анжелика','(+37529)1899545','г.Солигорск,ул.Набережная 24'), ('Соловей Дарья','(+37529)1243548','г.Слуцк,ул.Ленина 4'), ('Иван Иваныч','(+37529)145789623','г.Минск');

create table Zakaz(

Номер int, Дата\_сделки date, Наименование\_товара nvarchar(50) FOREIGN KEY REFERENCES TOVAR(Наименование\_товара), Покупатель nvarchar(50) FOREIGN KEY REFERENCES Zakazchiki(Покупатель), Кол\_заказанного\_товара int );

insert into Zakaz(Номер,Дата\_сделки,Наименование\_товара,Покупатель,Кол\_заказанного\_товара)

values (1,'02.03.2016','Зеркало','Слепцова Анжелика',2), (2,'10.02.2016','Столешница','Соловей Дарья',1);

-- drop table Zakaz

select \* FROM Tovar;

select Наименование\_товара,Цена from Tovar;

select Наименование\_товара from Zakaz Where Дата\_сделки = '02.03.2016';

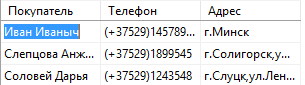
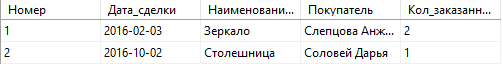
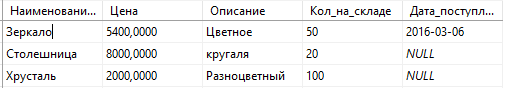
alter table Tovar add Дата\_поступления date;

update Tovar set Дата\_поступления = '03.06.2016' where Наименование\_товара='Зеркало';

drop table Tovar;

drop table Zakazchiki;

drop table Zakaz;



**Лабораторная работа №4. Настройка базы данных при ее создании**

1. Ознакомиться с приложением и создать базу данных ТМРX\_BSTU с помощью скрипта, где Х – первые буквы своей фамилии.
2. Разработать сценарий на языке T-SQL для создания и заполнения таблиц PROFESSION, PUL-PIT, TEACHER, SUBJECT, AUDITORIUM, GROUPS, STUDENT базы данных ТМРX\_BSTU.

Использовать нужные ограничения целостности из тех, что приведены в таблице справа.

Выполнить сценарий. Убедиться, что сценарий работает корректно.

Сохранить сценарий под именем CreateX\_BSTU.

1. Разработать сценарий для создания таблиц FACULTY и AUDITORIUM\_TYPE.

Выполнить сценарий, в случае необходимости откорректировать его и перенести операторы создания таблиц FACULTY и AUDITORIUM\_ TYPE в сценарий п. 2.

Сохранить сценарий под именем CreateX\_BSTU.

Удалить базу данных ТМРX\_BSTU.

1. Внести изменения в сценарий CreateX\_BSTU для создания базы данных X\_BSTU (где Х  первые буквы своей фамилии) в соответствии с приложением и характеристиками, приведенными в таблице ниже.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя базы данных | | | X\_BSTU |
| Модель восстановления | | | Простая (SIMPLE) |
| Кол-во файловых групп | | | 3 |
| Имена файловых групп | | | PRIMARY, G1, G2(по умолчанию) |
| Файлы группы PRIMARY | | | |
| X\_BSTU\_mdf | расположение: C:\BSTU\ X\_BSTU.mdf | | |
| начальный размер: 5 Mb | | |
| максимальный размер: 10 Mb | | |
| приращение: 1 Mb | | |
| X\_BSTU\_ndf | расположение: C:\BSTU\ X\_BSTU.ndf | | |
| начальный размер: 5 Mb | | |
| максимальный размер: 10 Mb | | |
| приращение: 10% | | |
| Файлы группы G1 | | | |
| X\_BSTU11\_ndf | | расположение:C:\BSTU\ Х\_BSTU11.ndf | |
| начальный размер: 10 Mb | |
| максимальный размер: 15 Mb | |
| приращение: 1 Mb | |
| X\_BSTU12\_ndf | | расположение: C:\BSTU\X\_BSTU12.ndf | |
| начальный размер: 2 Mb | |
| максимальный размер: 5 Mb | |
| приращение: 1 Mb | |
| Файлы группы G2 | | | |
| X\_BSTU21\_ndf | | расположение: C:\BSTU\X\_BSTU21.ndf | |
| начальный размер: 5 Mb | |
| максимальный размер: 10 Mb | |
| приращение: 1 Mb | |
| X\_BSTU22\_ndf | | расположение: C:\BSTU\X\_BSTU22.ndf | |
| начальный размер: 2 Mb | |
| максимальный размер: 5 Mb | |
| приращение: 1 Mb | |
| Журнал транзакций | | | |
| X\_BSTU\_log | | расположение: C:\BSTU\ X\_BSTU.ldf | |
| начальный размер: 5 Mb | |
| максимальный размер: не ограничен | |
| приращение: 1 Mb | |

1. Откорректировать сценарий п. 4 таким образом, чтобы созданные таблицы располагались в файловых группах в соответствии с таблицей ниже.

|  |  |
| --- | --- |
| Файловая группа | Таблица |
| PRIMARY | FACULTY |
| AUDITORIUM\_TYPE |
| PROFESSION |
| G1 | PULPIT |
| SUBJECT |
| GROUPS |
| TEACHER |
| G2 | AUDITORIUM |

1. Выполнить сценарий создания базы данных X\_BSTU. Убедиться, что база данных X\_BSTU создана, характеристики ее соответствуют заданным, таблицы расположены в нужных файловых группах и заполнены данными в соответствии с таблицами приложения.
2. С помощью SSMS построить диаграмму БД X\_BSTU и убедиться, что все требуемые внешние ключи (FORIGN KEY) отражены на диаграмме.

use master;

create database BSTU

on primary

( name = N'BSTU\_mdf', filename = N'D:\BD\BSTU\_mdf.mdf',

size = 10240Kb, maxsize=UNLIMITED, filegrowth=1024Kb),

( name = N'BSTU\_ndf', filename = N'D:\BD\BSTU\_ndf.ndf',

size = 10240KB, maxsize=1Gb, filegrowth=25%),

filegroup FG1

( name = N'BSTU\_fg1\_1', filename = N'D:\BD\BSTU\_fgq-1-1.ndf',

size = 10240Kb, maxsize=1Gb, filegrowth=25%),

( name = N'BSTU\_fg1\_2', filename = N'D:\BD\BSTU\_fgq-1-2.ndf',

size = 10240Kb, maxsize=1Gb, filegrowth=25%),

filegroup FG2

( name = N'BSTU\_fg2\_1', filename = N'D:\BD\BSTU\_fgq-2-1.ndf',

size = 10240Kb, maxsize=1Gb, filegrowth=25%),

( name = N'BSTU\_fg2\_2', filename = N'D:\BD\BSTU\_fgq-2-2.ndf',

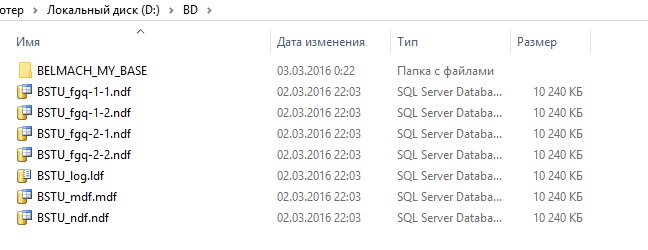
size = 10240Kb, maxsize=1Gb, filegrowth=25%)

log on

( name = N'BSTU\_log', filename=N'D:\BD\BSTU\_log.ldf',

size=10240Kb, maxsize=2048Gb, filegrowth=10%)

go



USE master;

CREATE DATABASE BSTU;

USE master

go

CREATE DATABASE BSTU

ON PRIMARY

(

name = N'BSTU\_mdf',

FILENAME = N'D:\University\Databases\Laboratory work\BSTU.mdf',

size = 5Mb,

maxsize = 10Mb,

filegrowth = 1Mb

),

(

name = N'BSTU\_ndf',

FILENAME = N'D:\University\Databases\Laboratory work\BSTU.ndf',

size = 5Mb,

maxsize = 10Mb,

filegrowth = 10%

),

FILEGROUP G1

(

name = N'BSTU\_G1\_1\_ndf',

FILENAME = N'D:\University\Databases\Laboratory work\BSTU\_G1\_1.ndf',

size = 10Mb,

maxsize = 15Mb,

filegrowth = 1Mb

),

(

name = N'BASE\_G1\_2\_ndf',

FILENAME = N'D:\University\Databases\Laboratory work\BSTU\_G1\_2.ndf',

size = 2Mb,

maxsize = 5Mb,

filegrowth = 1Mb

),

FILEGROUP G2

(

name = N'BSTU\_G2\_1',

FILENAME = N'D:\University\Databases\Laboratory work\BSTU\_G2\_1.ndf',

size = 5Mb,

maxsize = 10Mb,

filegrowth = 1Mb

),

(

name = N'BSTU\_G2\_2',

FILENAME = N'D:\University\Databases\Laboratory work\BSTU\_G2\_2.ndf',

size = 2Mb,

maxsize = 5Mb,

filegrowth = 1Mb

)

LOG ON

(

name = N'BSTU\_log',

FILENAME = N'D:\University\Databases\Laboratory work\BSTU\_ldf.ldf',

size = 5Mb,

maxsize = UNLIMITED,

filegrowth = 1Mb

)

go

USE BSTU;

/\*CREATE\*/

CREATE TABLE Faculty

(

Faculty char(10) PRIMARY KEY NOT NULL, /\*код факультета\*/

Faculty\_name varchar(50) DEFAULT '???' /\*наименование факультета\*/

);

CREATE TABLE Profession

(

Profession char(20) PRIMARY KEY NOT NULL, /\*код специальности\*/

Faculty char(10) FOREIGN KEY REFERENCES Faculty(Faculty) NOT NULL, /\*код факультета\*/

Profession\_name varchar(100) NULL, /\*наименование специальности\*/

Qualification varchar(50) NULL /\*квалификация\*/

);

CREATE TABLE Pulpit

(

Pulpit char(20) PRIMARY KEY NOT NUll, /\*код кафедры\*/

Pulpit\_name varchar(100) NULL, /\*наименование кафедры\*/

Faculty char(10) FOREIGN KEY REFERENCES Faculty(Faculty) NOT NULL /\*код факультета\*/

) ON G1;

CREATE TABLE Teacher

(

Teacher char(10) NOT NULL, /\*код преподавателя\*/

Teacher\_name varchar(100) NULL, /\*фамилия, имя, отчество преподавателя\*/

Gender char(1) CHECK (Gender in ('м', 'ж')), /\*пол\*/

Pulpit char(20) FOREIGN KEY REFERENCES Pulpit(Pulpit) NOT NULL /\*код кафедры\*/

) ON G1;

CREATE TABLE Subjects

(

Subjects char(10) PRIMARY KEY NOT NULL, /\*код дисциплины\*/

Subjects\_name varchar(100) NULL UNIQUE, /\*наименование дисциплины\*/

Pulpit char(20) FOREIGN KEY REFERENCES Pulpit(Pulpit) NOT NULL /\*код кафедры\*/

) ON G1;

CREATE TABLE Auditorium\_type

(

Auditorium\_type char(10) PRIMARY KEY NOT NULL, /\*код типа аудитории\*/

Auditorium\_type\_name varchar(30) NULL /\*нименование типа удитории\*/

);

CREATE TABLE Auditorium

(

Auditorium char(20) PRIMARY KEY NOT NULL, /\*код аудитории\*/

Auditorium\_type char(10) FOREIGN KEY REFERENCES Auditorium\_type(Auditorium\_type) NOT NULL, /\*код типа аудитории\*/

Auditorium\_capacity int DEFAULT 1 CHECK (Auditorium\_capacity BETWEEN 1 and 300), /\*вместимость\*/

Auditorium\_name varchar(50) NULL /\*наименование\*/

) ON G2;

CREATE TABLE Groups

(

ID\_group int PRIMARY KEY NOT NULL, /\*идентификатор группы\*/

Faculty char(10) FOREIGN KEY REFERENCES Faculty(Faculty) NOT NULL, /\*код факультета\*/

Profession char(20) FOREIGN KEY REFERENCES Profession(Profession) NOT NULL, /\*код специальности\*/

Year\_First smallint CHECK (Year\_First < YEAR(GETDATE())), /\*год поступления\*/

/\*вычисляемое поле "курс"\*/

) ON G1;

CREATE TABLE Students

(

ID\_student int PRIMARY KEY IDENTITY(1000, 1), /\*код студента\*/

ID\_group int FOREIGN KEY REFERENCES Groups(ID\_group) NOT NULL, /\*идентификатор группы\*/

Name nvarchar(100), /\*фамилия, имя, отчество студента\*/

Birthday date, /\*дата рождения\*/

Stamp timestamp, /\*штамп времени\*/

Info xml, /\*дополнительная информация\*/

Photo varbinary(max) /\*фотография\*/

);

CREATE TABLE Progress

(

Subjects char(10) FOREIGN KEY REFERENCES Subjects(Subjects), /\*предмет\*/

ID\_student int FOREIGN KEY REFERENCES Students(ID\_student), /\*код студента\*/

Pdate date, /\*дата экзамена\*/

Note int CHECK (Note BETWEEN 1 and 10) /\*оценка\*/

);

/\*INSERT\*/

INSERT INTO Faculty

VALUES ('ТТЛП', 'Технологии и техника лесной промышленности'),

('ТОВ', 'Технологии органических веществ'),

('ХТиТ', 'Химические технологии и техника'),

('ИЭФ', 'Инженерно-экономический'),

('ЛХ', 'Лесохозяйственный'),

('ИДиП', 'Издательское дело и полиграфия'),

('ИТ', 'Информационных технологий');

INSERT INTO Profession

VALUES ('1-36 06 01', 'ИДиП', 'Полиграфическое оборудование и системы обработки информации', 'инженер-электромеханиик'),

('1-36 07 01', 'ХТиТ', 'Машины и аппараты химических производств и предприятий строительных материалов', 'инженер-механик'),

('1-40 01 02', 'ИДиП', 'Информационные системы и технологии', 'инженер-программист-системотехник'),

('1-46 01 01', 'ТТЛП', 'Лесоинженерное дело', 'инженер-технолог'),

('1-47 01 01', 'ИДиП', 'Издательское дело', 'редактор-технолог'),

('1-48 01 02', 'ТОВ', 'Химическая технология органических веществ, материалов и изделий', 'инженер-химик-технолог'),

('1-48 01 05', 'ТОВ', 'Химическая технология переработки древисины', 'инженер-химик-технолог'),

('1-54 01 03', 'ТОВ', 'Физико-химические методы и приборы контроля качества продукции', 'инженер по сертификации'),

('1-75 01 01', 'ЛХ', 'Лесное хозяйство', 'инженер лесного хозяйства'),

('1-75 02 01', 'ЛХ', 'Садово-парковое строительство', 'инженер садово-паркового строительства'),

('1-89 02 02', 'ЛХ', 'Туризм и природопользование', 'специалист в сфере туризма');

INSERT INTO Pulpit

VALUES ('РИТ', 'Редакционно-издательских технологий', 'ИДиП'),

('СБУАиА', 'Статистики, бухгалтерского учета, анализа и аудита', 'ИЭФ'),

('ТДП', 'Технологий деревообрабатывающих производств', 'ТТЛП'),

('ТиДИД', 'Технологий и дизайна изделий из древесины', 'ТТЛП'),

('ТиП', 'Туризма и природопользования', 'ЛХ'),

('ТЛ', 'Тренспорта леса', 'ТТЛП'),

('ТНВиОХТ', 'Технологий неорганических веществ и общей химической технологии', 'ХТиТ'),

('ТНХСиППМ', 'Технологий нефтехимического синтеза и переработки полимерных материалов', 'ТОВ'),

('ХПД', 'Химической переработки древесины', 'ТОВ'),

('ХТЭПиМЭЕ', 'Химии, технологий электрохимических производств и материалов электронной техники', 'ХТиТ'),

('ЭТиМ', 'Экономической теории и маркетинга', 'ИЭФ');

INSERT INTO Pulpit

VALUES ('ИСиТ', 'Информационных систем и технологий', 'ИТ'),

('ПОиСОИ', 'Полиграфическое оборудование и системы обработки информации', 'ИДиП');

INSERT INTO Teacher (Teacher, Teacher\_name, Pulpit)

VALUES ('НСКВ', 'Носков Михаил Трофимович', 'ТДП'),

('ПРКП', 'Прокопенко Николай Иванович', 'ТНХСиППМ'),

('МРЗВ', 'Морозова Елена Степановна', 'ИСиТ'),

('РВКС', 'Ровкас Андрей Петрович', 'ЭТиМ'),

('РЖКВ', 'Рыжиков Леонид Николаевич', 'ТЛ'),

('РМНВ', 'Романов Дмитрий Михайлович', 'ИСиТ'),

('СМЛВ', 'Смелов Владимир Владилавович', 'ИСиТ'),

('КРЛВ', 'Крылов Павел Павлович', 'ИСиТ'),

('ЧРН', 'Чернова Анна Викторовна', 'ХПД'),

('МХВ', 'Мохов Михаил Сергеевич', 'ПОиСОИ');

INSERT INTO Subjects

VALUES ('ПЗ', 'Представление знаний в компьютерных системах', 'ИСиТ'),

('ПИС', 'Проектирование информационных систем', 'ИСиТ'),

('ПМАПЛ', 'Полиграфические машины, автоматы и поточные линии', 'ПОиСОИ'),

('ПСП', 'Программирование сетевых приложений', 'ИСиТ'),

('ПЭХ', 'Прикладная электрохимия', 'ХТЭПиМЭЕ'),

('СУБД', 'Системы управления базами данных', 'ИСиТ'),

('ТиОЛ', 'Технология и оборудование лесозаготовок', 'ТЛ'),

('ТОПИ', 'Технология обогащения полезных ископаемых', 'ТНВиОХТ'),

('ТРИ', 'Технология резиновых изделий', 'ТНХСиППМ'),

('ЭП', 'Экономика прородопользования', 'ТиП'),

('ЭТ', 'Экономическая теория', 'ЭТиМ');

INSERT INTO Auditorium\_type

VALUES ('ЛБ-Х', 'Химическая лаборатория'),

('ЛБ-К', 'Компьютерный класс'),

('ЛБ-СК', 'Специальный компьютерный класс'),

('ЛК', 'Лекционная аудитория'),

('ЛК-К', 'Лекционная ауд. с проектором');

INSERT INTO Auditorium

VALUES ('301-1', 'ЛБ-К', 15, '301-1'),

('304-4', 'ЛБ-К', 90, '304-4'),

('305-4', 'ЛК-К', 90, '305-4'),

('313-1', 'ЛК-К', 60, '313-1'),

('314-4', 'ЛК', 90, '314-4'),

('320-4', 'ЛК', 90, '320-4'),

('324-1', 'ЛК-К', 50, '324-1'),

('408-2', 'ЛК', 90, '408-2'),

('413-1', 'ЛБ-К', 15, '413-1'),

('423-1', 'ЛБ-К', 90, '423-1'),

('429-4', 'ЛК', 90, '429-4');

INSERT INTO Groups

VALUES (22, 'ЛХ', '1-75 02 01', 2011),

(23, 'ЛХ', '1-89 02 02', 2012),

(24, 'ЛХ', '1-89 02 02', 2011),

(25, 'ТТЛП', '1-46 01 01', 2013),

(26, 'ТТЛП', '1-46 01 01', 2012),

(27, 'ТТЛП', '1-46 01 01', 2012),

(28, 'ИДиП', '1-36 06 01', 2013),

(29, 'ИДиП', '1-36 06 01', 2012),

(30, 'ИДиП', '1-36 06 01', 2010),

(31, 'ИДиП', '1-47 01 01', 2013),

(32, 'ИДиП', '1-40 01 02', 2012);

INSERT INTO Students (ID\_group, Name, Birthday)

VALUES (22, 'Пугач Михаил Трофимович', '12.01.1996'),

(23, 'Авдеев Николай Иванович', '19.07.1996'),

(24, 'Белова Елена Станиславовна', '22.05.1996'),

(25, 'Вилков Анрей Петрович', '08.12.1996'),

(26, 'Грушин Леонид Николаевич', '11.11.1995'),

(27, 'Дунаев Дмитрий Михайлович', '24.08.1996'),

(28, 'Клуни Иван Владиславович', '15.09.1996'),

(29, 'Крылов Олег Павлович', '16.10.1996'),

(30, 'Петрова Анна Викторовна', '27.03.1996'),

(31, 'Минич Михаил Сергеевич', '28.04.1996'),

(32, 'Зорин Виктор Степанович', '30.05.1996');

INSERT INTO Progress

VALUES ('ЭТ', 1000, '12.01.2014', 4),

('ЭТ', 1001, '19.01.2014', 5),

('ЭТ', 1002, '22.01.2014', 6),

('ЭТ', 1003, '08.01.2014', 9),

('ПЗ', 1004, '11.01.2014', 8),

('ПЗ', 1005, '24.01.2014', 7),

('ПЗ', 1006, '15.01.2014', 4),

('СУБД', 1007, '16.01.2014', 7),

('СУБД', 1008, '27.01.2014', 6),

('СУБД', 1009, '28.01.2014', 5),

('ТРИ', 1010, '30.01.2014', 4);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USE MASTER;

CREATE database MY\_Prodagi

on primary

(name =N'PRODAGI\_mdf',

filename=N'e:\базы данных\PRODAGI\_mdf.mdf',

size = 5Mb,

maxsize=10Mb,

filegrowth=1Mb),

(NAME =N'PRODAGI\_ndf',

filename=N'e:\базы данных\PRODAGI\_ndf.ndf',

size = 5Mb,

maxsize=10Mb,

filegrowth=10%),

FILEGROUP P1

(NAME=N'PRODAGI\_g1\_1',

filename=N'e:\базы данных\PRODAGI\_g-1.ndf',

size = 10Mb,

maxsize=15Mb,

filegrowth=1Mb),

(NAME=N'PRODAGI\_g2\_2',

filename=N'e:\базы данных\PRODAGI\_g-2.ndf',

size = 2Mb,

maxsize=5Mb,

filegrowth=1Mb),

FILEGROUP P2

(NAME=N'PRODAGI\_2g\_1',

filename=N'e:\базы данных\PRODAGI\_g-21.ndf',

size = 5Mb,

maxsize=10Mb,

filegrowth=1Mb),

(NAME=N'PRODAGI\_2g\_2',

filename=N'e:\базы данных\PRODAGI\_g-22.ndf',

size = 2Mb,

maxsize=5Mb,

filegrowth=1Mb)

log on

( name = N'PRODAGI\_log',

filename=N'e:\базы данных\PRODAGI\_log.ldf',

size=5Mb,

maxsize=UNLIMITED,

filegrowth=1Mb)

go

use MY\_Prodagi;

create table Tovar(

Наименование\_товара nvarchar(50) primary key not null, Цена money, Описание nvarchar(50), Кол\_на\_складе int)ON P1;

insert into Tovar(Наименование\_товара,Цена,Описание,Кол\_на\_складе)

values ('Зеркало',5400,'Цветное',50), ('Хрусталь',2000,'Разноцветный', 100), ('Столешница', 8000,'кругаля',20);

create table Zakazchiki(

Покупатель nvarchar(50) primary key not null, Телефон nvarchar(50), Адрес nvarchar(50))ON P2;

insert into Zakazchiki(Покупатель,Телефон,Адрес)

values ('Слепцова Анжелика','(+37529)1899545','г.Солигорск,ул.Набережная 24'), ('Соловей Дарья','(+37529)1243548','г.Слуцк,ул.Ленина 4'), ('Иван Иваныч','(+37529)145789623','г.Минск');

create table Zakaz(

Номер int, Дата\_сделки date, Наименование\_товара nvarchar(50) FOREIGN KEY REFERENCES TOVAR(Наименование\_товара), Покупатель nvarchar(50) FOREIGN KEY REFERENCES Zakazchiki(Покупатель), Кол\_заказанного\_товара int )ON P2;

insert into Zakaz(Номер,Дата\_сделки,Наименование\_товара,Покупатель,Кол\_заказанного\_товара)

values (1,'02.03.2016','Зеркало','Слепцова Анжелика',2), (2,'10.02.2016','Столешница','Соловей Дарья',1);

-- drop table Zakaz

SELECT \* FROM Tovar;

SELECT Наименование\_товара,Цена from Tovar;

SELECT Наименование\_товара from Zakaz Where Дата\_сделки = '02.03.2016';

drop table Tovar;

drop table Zakazchiki;

drop table Zakaz;

