

Лабораторная работа

Физическое проектирование и создание базы данных в MS SQL Server

Общая постановка задачи: Создайте базу данных **ФАКУЛЬТЕТ**, которая, аналогично задаче, поставленной в **лабораторной работе №4**. Данная БД будет содержать сведения о факультет некоего ВУЗа (см. задания 4 и 5, а также стр. 3-4 в ЛР №4).

Задание 1.

А) Запустите компонент SQL Server Management Studio и произведите соединение с сервером (параметры соединения см. в ЛР №1).

Б) Создайте новую базу данных **ФАКУЛЬТЕТ_ВашаФамилия** (см. ЛР №4).

- **Имя базы данных:** **ФАКУЛЬТЕТ_ВашаФамилия**;
- **Владелец:** [sa] и [VYATSU\CLASS];
- **Файлы базы данных / Путь** для обоих файлов БД папка «БД_ФАКУЛЬТЕТ» на внешнем носителе (Вашей флешке), предварительно такую папку нужно создать или папка C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA.

Создание, изменение и сохранение таблиц БД в Обозревателе объектов

Создание таблиц БД

Создать новую таблицу можно в **Обозревателе объектов** либо в **диаграмме базы данных** (о ней чуть позже).

В первом случае в **Обозревателе объектов** в контекстном меню раздела **Таблицы** базы данных вызывается команда **Создать ► Таблица** (рис. 1).

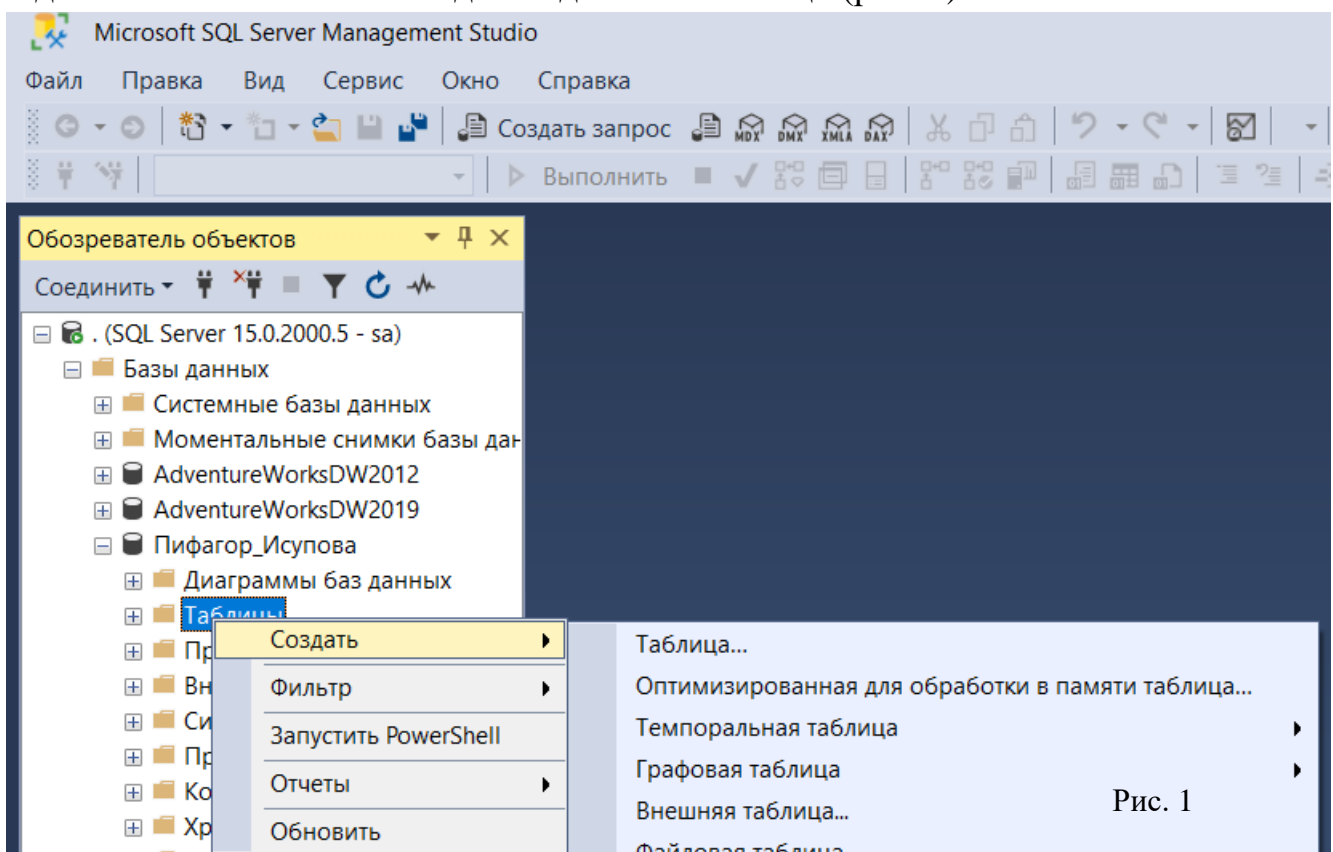
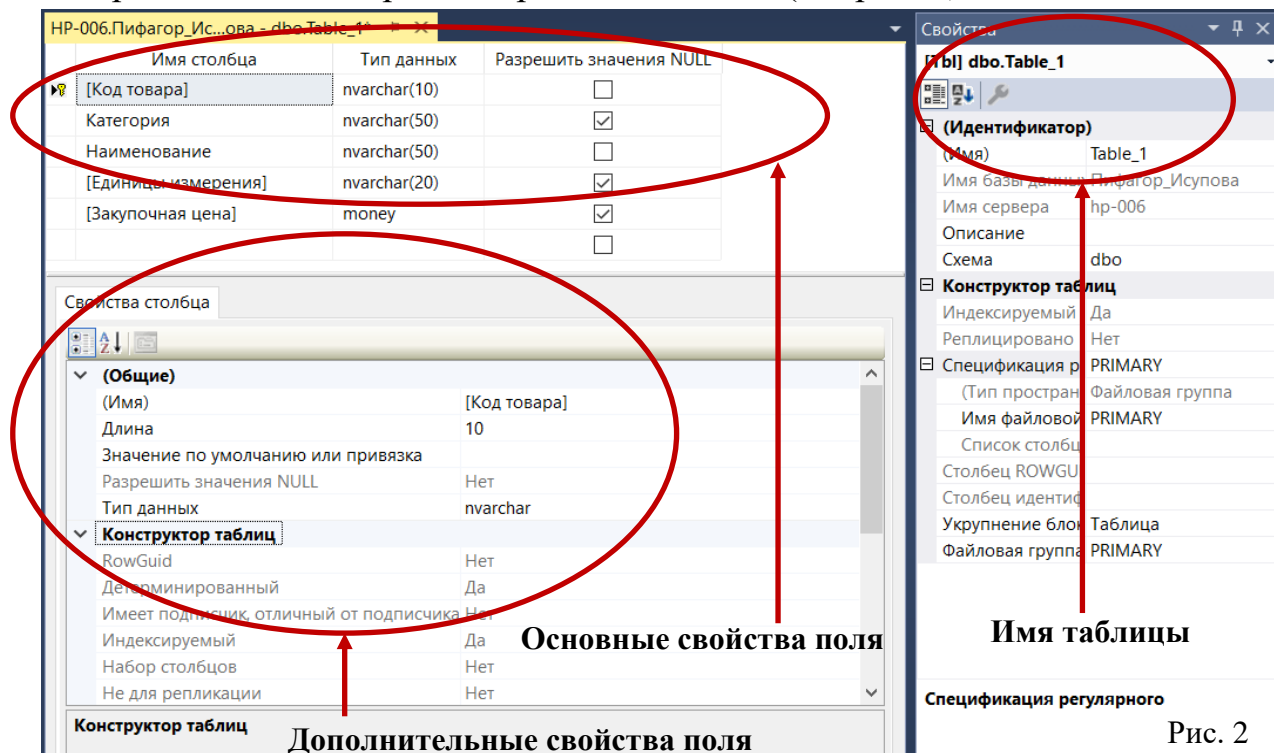


Рис. 1

В центральной части экрана откроется табличка (см. рис. 2).



Имя таблице задается при сохранении или может быть установлено сразу в правой части окна на панели свойств всей таблицы, вызывается командой **Свойства** или **ALT+Enter** (см. рис. 2).

В **основных свойствах** задаются поля таблицы с указанием типов и обязательности / необязательности значений, а также первичный ключ. Если значение поля обязательно должно быть внесено, то указывается недопустимость Null-значений (т.е. пустых значений), если поле можно оставить пустым – указывается допустимость Null-значений.



Основные типы данных

Название	Описание
char (n) nchar (n)	Строка фиксированной длины, где <i>n</i> – длина строки. Значения вводятся в апострофах
varchar (n) nvarchar (n)	Строка переменной длины, где <i>n</i> – максимально возможная ее длина. Значения вводятся в апострофах
int	Целое число со знаком размером 4 байта
smallint	Короткое целое число со знаком размером 2 байта
decimal (n, m)	Вещественное число, где <i>n</i> – общая длина числа, <i>m</i> – количество знаков справа от десятичной точки
money	Денежный тип данных размером 8 байт
smallmoney	Короткий денежный тип данных размером 4 байта
datetime	Дата и время
bit	Целочисленный тип данных, который может принимать значения 1, 0 или NULL

Кроме того, уточнить **дополнительные свойства столбца (поля)** можно в отдельной области внизу окна (см. рис. 2): значение по умолчанию или привязка; спецификация вычисляемого столбца (Формула) и др.


Задание первичного ключа

Назначить поле **первичным ключом** можно:

- через команду  **Задать первичный ключ** контекстного меню этого поля;
- командой  **Задать первичный ключ** на панели инструментов;
- через команду **Задать первичный ключ** в меню **Конструктор таблиц**.

Задание ограничений

Простые ограничения можно задать в дополнительных свойствах поля, например, в свойстве «**Значение по умолчанию или привязка**» (рис. 3).

Ограничения в виде выражений задаются в специальном окне **Проверочные ограничения**, которое открывается через контекстное меню, вызванное на имени поля, или через меню **Конструктор таблиц** (команда  **Проверочные ограничения...**). В открывшемся окне новое ограничение вносится кнопкой **Добавить** (рис. 4), затем в строке **Выражение** вносится соответствующее выражение (правило проверки), в строке **(Имя)** желательно задать смысловое имя для ограничения, чтобы в дальнейшем было удобнее с ним работать.

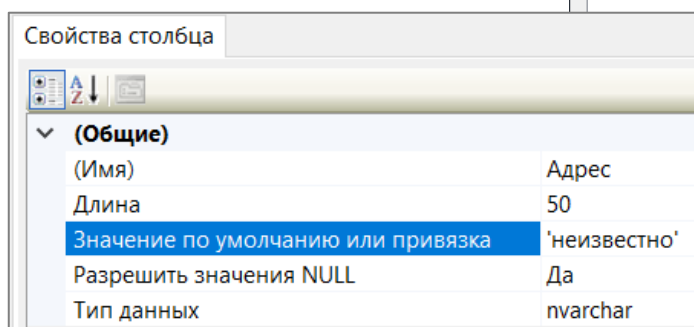


Рис. 3

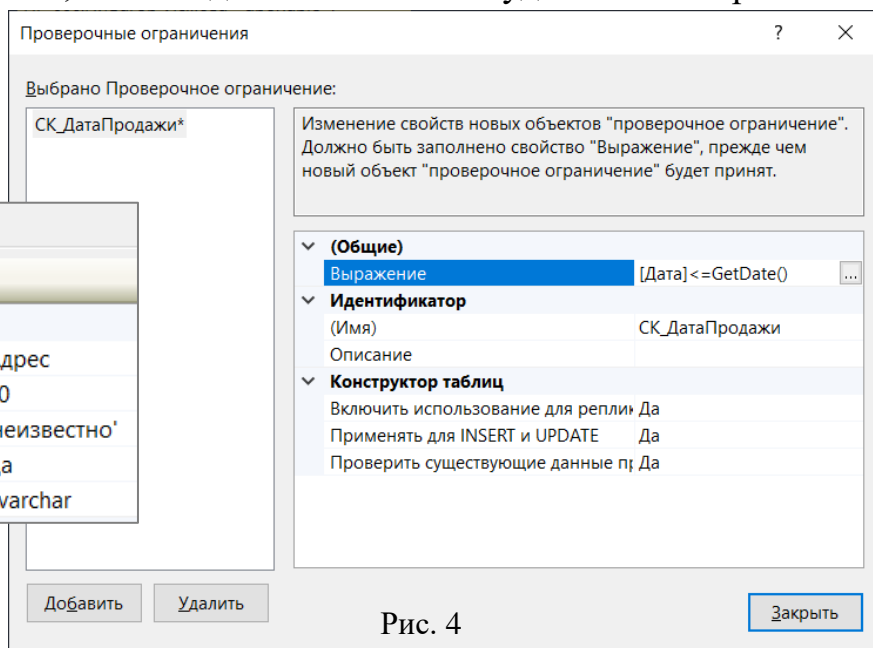




Рис. 4

Изменение таблиц БД

Изменение структуры таблицы или свойств ее полей производится аналогично ее созданию (команда **Проект** в контекстном меню таблицы в **Обозревателе объектов**).

Сохранение таблиц БД

Сохранить таблицу или произведенные изменения можно:

- через команду **Сохранить ...** контекстного меню на вкладке таблицы;
- командами  **Сохранить** и  **Сохранить все** на панели инструментов;
- сочетаниями клавиш **Ctrl+S** или **Ctrl+Shift+S**.

Если при сохранении изменений в таблице возникнет ошибка, связанная с запретом сохранения изменений, требующих повторного создания таблицы, то нужно выполнить команду **Параметры...** в меню **Сервис**. В появившемся окне выбрать слева раздел

Конструкторы и снять флажок около параметра «Запретить сохранение изменений, требующих повторного создания таблицы» (рис. 5):

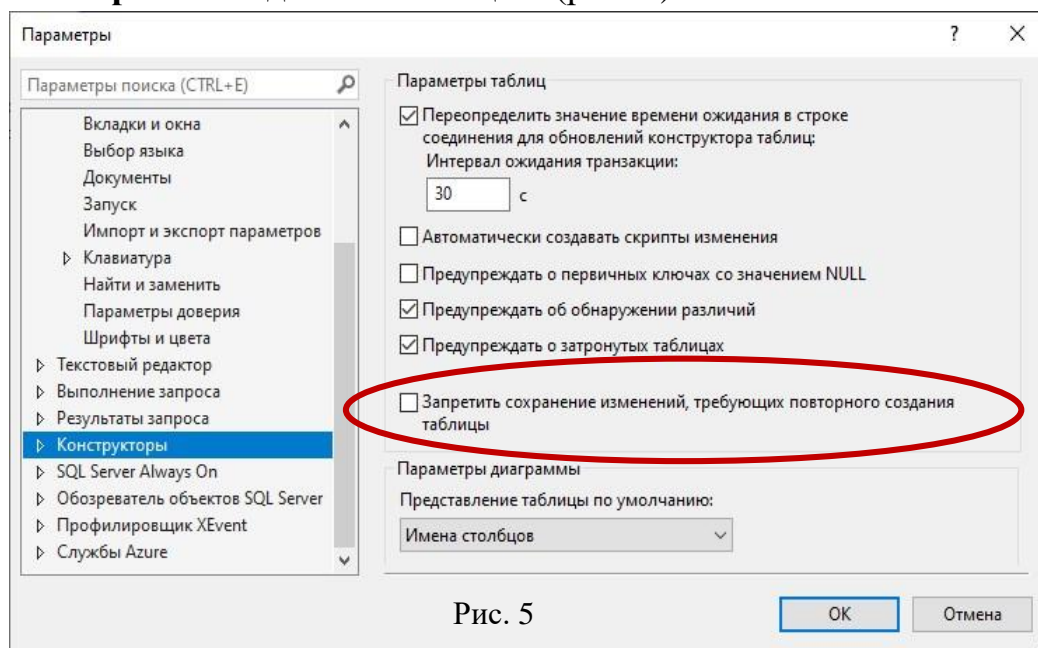


Рис. 5

Задание 2. В своей базе данных ФАКУЛЬТЕТ через **Обозреватель объектов** создайте **четыре** таблицы **Группа**, **Студент**, **Преподаватель**, **Дисциплина** по описанию, приведенному в **ЛР №4** на стр. 3–4. **Для всех полей согласно описанию, задайте:**

- соответствующие типы данных;
- обязательность / необязательность заполнения (значения NULL)
- первичные ключи;
- дополнительные ограничения: значения по умолчанию, проверочные ограничения, формулы для вычисляемых полей.

Примечания:

1) в этой версии языка SQL (Transact-SQL) функция, возвращающая текущую дату `GetDate()`

2) маску ввода и поля со списками здесь задавать нельзя, можно лишь через проверочные ограничения предусмотреть возможность ввода конкретных символов в определенном виде. Например, для полей «телефон» или «пол» (рис. 6).

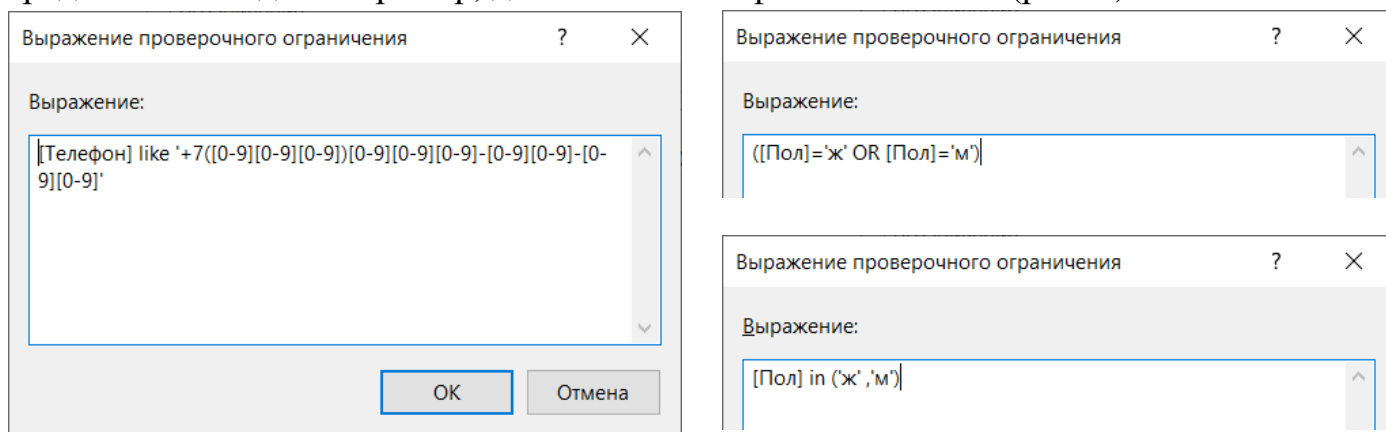
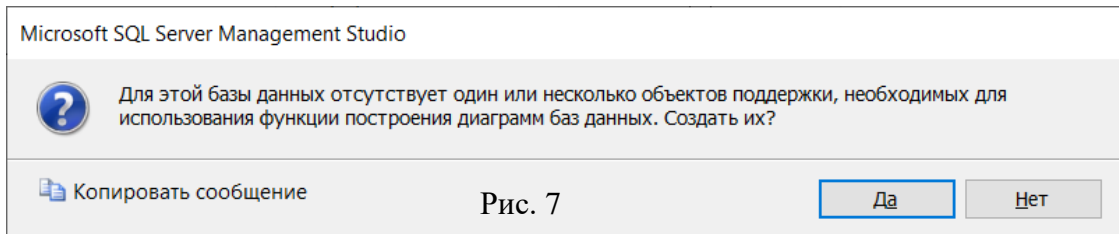


Рис. 6

Представление базы данных в виде Диаграммы

В SQL-Server есть средства для визуального отображения диаграммы базы данных. Для ее построения нужно в **Обозревателе объектов** в **Management Studio** выбрать нужную базу данных. Для нее на узле **Диаграммы баз данных** из контекстного меню выбрать пункт **Создать диаграмму базы данных**.

При первом создании Диаграммы возможно появление окна (рис. 7), в котором следует нажать «Да».



В открывшемся окне **Добавление таблицы** необходимо выделить те таблицы, которые будут помещены в диаграмму, и нажать кнопку **Добавить**. Затем в течение некоторого времени будут добавляться таблицы, после чего можно закрыть окно **Добавление таблицы**.

Задание 3. Постройте диаграмму своей базы данных ФАКУЛЬТЕТ.

Создание таблиц в Диаграмме базы данных


Создать таблицу можно в окне **диаграммы БД** с помощью команды контекстного меню  **Создать таблицу...** (рис. 8).



Рис. 8

Можно изменить вид таблицы в диаграмме – в контекстном меню таблицы выбрать пункт **Представления** (например, вид представления **Имена столбцов** – традиционный для схемы вид таблицы: только имена столбцов без указания их свойств, а вид представления **Стандартное** – со свойствами). Пока **около имени таблицы** стоит **символ ***, это означает, что изменения не были сохранены. Поэтому после определения всех необходимых параметров таблицы надо ее сохранить (а точнее, сохранить диаграмму с новой таблицей) – можно с помощью кнопки с изображением дискеты.

Свойства таблиц в целом можно задать через окно **Свойства** таблицы (вызвать его можно через контекстное меню таблицы).

Кроме того, свойства таблицы можно просмотреть и изменить через окно **Свойства** при открытой **диаграмме БД** (вызывается через меню **Вид** или кнопку  на панели инструментов).

Задание 4. В базу данных **ФАКУЛЬТЕТ** через ее **диаграмму** добавьте оставшуюся таблицу **Занятие**.

Задайте составной первичный ключ, а также все необходимые свойства и ограничения согласно описанию (см. ЛР №4 или [стр. 8–9 данной ЛР](#)).

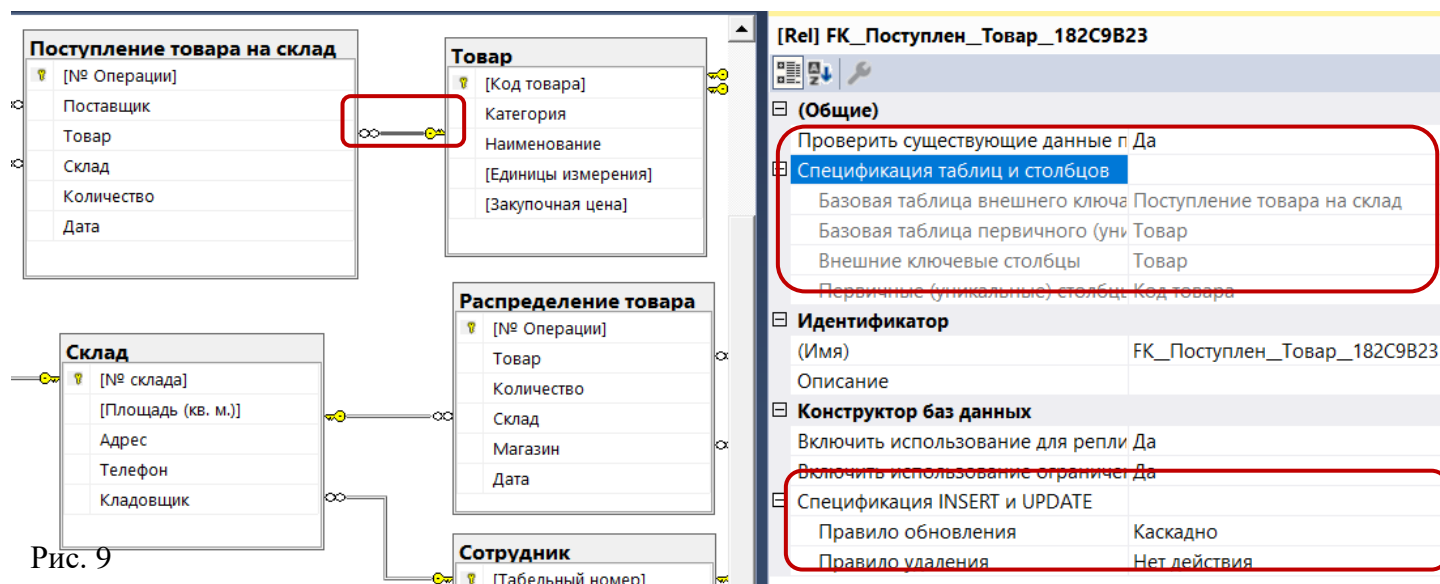
Сохраните диаграмму с новой таблицей.

Создание и изменение связей

Создание связей удобно производить в **диаграмме БД**. Можно просто «протянуть» мышью от одного связываемого поля к другому. Кроме того, новые связи можно описать через окно **Отношения...**, кнопка **Добавить**.

Просмотреть и изменить параметры связи можно:

- в окне **Свойства** (обычно в правой части экрана) на **диаграмме БД ПРИ ВЫДЕЛЕННОЙ СВЯЗИ** (рис. 9);
- в окне **Отношения внешнего ключа**, которое вызывается двойным щелчком на названии внешнего ключа в **Обзревателе объектов**.



Задание 5. Создайте все необходимые связи между таблицами базы данных **ФАКУЛЬТЕТ**, руководствуясь описанием связей в ЛР или логической моделью, построенной Вами в рамках ЛР №4.

Заполнение таблиц

Для любой БД важна последовательность заполнения таблиц – сначала лучше заполнить те таблицы, от первичных ключей которых идут связи. Либо временно не заполнять внешние ключи, если не внесены значения соответствующих первичных ключей (например, не заполнять поле **Заведующий** таблицы **Магазин**, пока не заполнена таблица **Сотрудник**).

Таблицы БД можно заполнять, выбрав команду **Изменить первые 200 строк** из контекстного меню определенной таблицы (рис. 10).

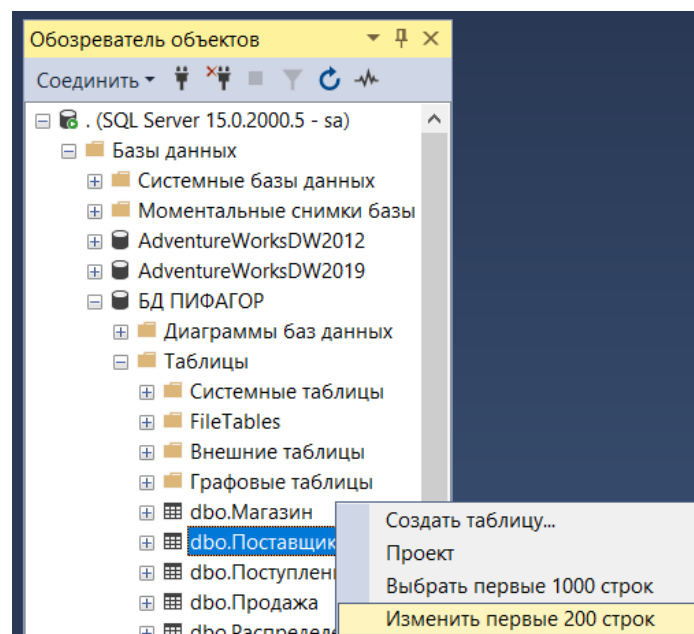


Рис. 10

Задание 6. Частично заполните БД ФАКУЛЬТЕТ: в каждую таблицу внесите по две-три записи.

Задание 7. Отсоедините свою БД ФАКУЛЬТЕТ, предварительно завершив все действия с БД, закрыв все ее объекты.

Задание 8. Заполните отчет о проделанной работе (файл «05_Отчет_ВашаФамилия»), прикрепите его в электронный курс «Базы данных» в Moodle.

Общая постановка задачи

На одном из факультетов некоторого учебного заведения (например, ВУЗа) проводятся учебные занятия.

Необходимо создать базу данных (БД), в которой будет вестись учет сведений об учебных занятиях: какие занятия ведутся, у каких учебных групп (в т. ч. учитывается состав группы), кто из преподавателей проводит занятия.

Логическая модель БД «ФАКУЛЬТЕТ»

Разработаем логическую структуру реляционной базы данных для решения поставленной задачи. Для этого обычно выполняются следующие шаги:

- 1 шаг – определение таблиц
- 2 шаг – определение полей таблиц
- 3 шаг – определение связей

Выделим следующие **характеристики (атрибуты)** для всех объектов, также укажем требования к атрибутам.

Группа

- *Код группы* – обязательный атрибут, первичный ключ;
- *Направление подготовки* – необязательный атрибут;
- *Профиль* – необязательный атрибут;
- *Курс* – необязательный атрибут.

Студент

- *Номер зачетной книжки* – обязательный атрибут, первичный ключ;
- *Фамилия, Имя* – обязательные атрибуты;
- *Отчество* – необязательный атрибут;
- *Пол* – необязательный атрибут, **предполагает ввод одного из двух возможных значений «ж» или «м»** (см. рис. 6);
- *Дата рождения* – необязательный атрибут, **не может быть больше сегодняшней даты и ранее 1900 года**;

Примечание: в версии языка SQL (Transact-SQL) функция, возвращающая текущую дату, – `GetDate()`, а функция, возвращающая год от даты рождения, выглядит так: `datepart(year,[Дата рождения])` или `Year([Дата рождения])`;

- *Адрес* – необязательный атрибут, если не указан, то **по умолчанию устанавливается значение «неизвестно»**;
- *Сотовый телефон* – необязательный атрибут, **предполагает ввод символов в формате: +7(000)000-00-00** (где 0 – любая цифра) (см. рис. 6);
- *E-mail* – необязательный атрибут;
- *Год поступления* – необязательный атрибут;
- *Проживает в общежитии* – обязательный атрибут.

Преподаватель

- *Код преподавателя* – обязательный атрибут, генерируемый автоматически;
- *Фамилия, Имя* – обязательные атрибуты;
- *Отчество* – необязательный атрибут;
- *Сотовый телефон* – необязательный атрибут, **предполагает ввод символов в формате: +7(000)000-00-00** (где 0 – любая цифра).
- *E-mail* – необязательный атрибут;
- *Рабочий телефон* – необязательный атрибут, формат ввода '00-00-00';
- *Кафедра* – обязательный атрибут;

- *Должность* – обязательный атрибут;
- *Ученая степень* – необязательный атрибут;
- *Дата принятия на работу* – обязательный атрибут, **не может быть больше текущей даты, по умолчанию устанавливается текущая дата;**
- *Дата рождения* – необязательный атрибут, **не может быть больше сегодняшней даты и ранее 1900 года.**

Дисциплина

- *Код дисциплины* – обязательный атрибут, первичный ключ;
- *Название* – обязательный атрибут;
- *Количество лекционных часов* – обязательный атрибут, целое **положительное** число;
- *Количество практических часов* – обязательный атрибут, целое **положительное** число;
- *Общий объем часов* – обязательный атрибут, **вычисляется по формуле** как сумма количества лекционных и практических часов.
- *Семестр изучения* – обязательный атрибут, **целое число от 1 до 10;**
- *Форма контроля* – обязательный атрибут, **предполагает ввод одного из двух возможных значений «зачет» или «экзамен».**

Занятие

- *Группа* – обязательный атрибут, входит в состав первичного ключа;
- *Дисциплина* – обязательные атрибуты, входит в состав первичного ключа;
- *Вид занятия* – обязательный атрибут, входит в состав первичного ключа;
- *Преподаватель* – обязательный атрибут.

Успеваемость

- *№Зачетки* – обязательный атрибут;
- *Дата* – обязательный атрибут;
- *Дисциплина* – обязательный атрибут;
- *Преподаватель* – обязательный атрибут;
- *Оценка* – обязательный атрибут.

Аудитория

- *№ Кабинета* – обязательный атрибут;
- *Количество мест* – обязательный атрибут, целое положительное число;
- *Компьютерный класс* – обязательный атрибут, **предполагает ввод одного из значений «да» или «нет», по умолчанию – «нет»;**
- *Проектор* – обязательный атрибут, **предполагает ввод одного из значений «да» или «нет», по умолчанию – «нет».**

При этом нужно учитывать следующие **ограничения (связи):**

- у каждой группы может проводиться несколько занятий;
- в каждой группе учится несколько студентов;
- каждый преподаватель может проводить несколько занятий;
- одна и та же дисциплина может проводиться разными преподавателями у разных групп и в различных формах (лекция, семинар, лабораторное занятие и т. п.);
- у каждого студента может быть несколько оценок по разным дисциплинам;
- занятие проводится в одном кабинете.