



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

КАФЕДРА «Вычислительные системы и информационная безопасность»

Методические рекомендации

по выполнению контрольной работы по дисциплине

«Хранение и защита информации в базах данных информационных систем»

г. Ростов-на-Дону

2022 г.

Задания: 1. Создать базу данных, содержащую данные о сотрудниках, подразделениях организации (в которой работает обучающийся) и статистическую информацию по ним.

1.1. Создать базу данных Microsoft SQL Server в MS Visual Studio содержащую данные о сотрудниках, подразделениях организации (в которой работает обучающийся) и статистическую информацию по ним..

1.2. В БД определить три новые таблицы со структурой полей.

1.3. Определить требуемые индексы для таблиц БД (первичный ключ и вторичные ключи). Установить связи между таблицами БД.

1.4. Внести в таблицы БД по 10-15 записей, содержание которых соответствует семантике и типу полей.

2. Создать хранимые процедуры, обеспечивающие возможность выполнения запросов к базе данных.

2.1 В отчете представить скриншоты хранимых процедур и результатов запуска процедур в виде таблицы

Описание процедуры	
SQL-конструкция для создания	Команда для извлечения
Результат запуска	

3. Администрирование SQL SERVER

3.1. Настройте параметры работы утилиты Enterprise Manager:

- установите интервал проверки текущего состояния SQL Server – 20 секунд, SQL Server Agent – 30 секунд, SQL Mail – 15

секунд, MS DistributedTransaction Coordinator – 60 секунд;

- настройте хранение информации о зарегистрированных экземплярах SQL Server в локальной системе реестра Windows, сделайте ее доступной другим локальным и удаленным пользователям.

Опишите, как Вы это сделали. Укажите назначение служб, интервал проверки по которым Вы настраивали.

3.2. Задайте дополнительные опции SQL Server:

- укажите время ожидания подключения к SQL Server равное 10 минутам;

- время ожидания выполнения запроса к удаленному серверу должно выполняться бесконечно долго;

- настройте Enterprise Manager таким образом, чтобы при открытии утилиты дерево объектов выводилось в том же виде, как и на момент закрытия утилиты.

Опишите, как Вы это сделали. Укажите назначение дополнительных опций.

3.3. Укажите, что регистрируется в журналах SQL Server Logs. Укажите как открыть текущий журнал. Приведите пример содержания журнала.

Объясните назначение представленных сообщений.

3.4. Используя утилиту Data Transformation Services (DTS), экспортируйте записи таблицы Employees базы данных Northwind в Txt-файл.

Приведите структуру таблицы Employee и экспортированные данные, представьте результатный Txt-файл.

3.5. Создайте свою базу данных, используя утилиту Enterprise Manager. Присвойте ей произвольное имя. Опишите, как Вы это сделали.

3.6. Скопируйте таблицу Products базы данных Northwind с помощью утилиты Data Transformation Services (DTS) в таблицу MyProduct Вашей базы данных.

Выберите только десять первых записей, используя запрос (Query) для спецификации данных. Опишите, как Вы это сделали.

3.7. Скопируйте всю базу данных Northwind в новую базу данных под именем Northwind_XXXX, где XXXX – номер Вашего студенческого билета. Воспользуйтесь для этого утилитой Data Transformation Services (DTS). Опишите, как Вы это сделали.

3.8. Почему представления базы данных Northwind превратились в таблицы Вашей базы данных? Приведите список этих представлений. Чем отличаются представления базы данных Northwind от одноименных таблиц Вашей базы данных? В чем их недостаток?

3.9. Для базы данных Northwind сгенерируйте ER-диаграмму (диаграмму сущность-связь) с помощью мастера Create Database Diagram Wizard. Используйте таблицы Categories, Products, Order Details, Suppliers, Orders, Shippers, Customers и Employees. Укажите, как Вы это сделали. Опишите созданные связи между таблицами. Какие для этого используются внешние ключи, по каким столбцам они построены?

3.10. Запустите утилиту Query Analyzer. Приведите перечень функций для работы с датами и временем. Воспользуйтесь системой оперативной помощи Books Online, укажите назначение и приведите примеры использования функций DateAdd и DatePart.

4. Управление серверами MS SQL SERVER

4.1. Удалите регистрационную информацию о сервере (LOCAL). Опишите, как Вы это сделали.

4.2. Создайте новую регистрационную информацию о Вашем сервере с помощью Register SQL Server Wizard или диалогового окна SQL Server Properties. Выберите любой из доступных серверов. Опишите, как Вы это сделали.

4.3. Создайте новую серверную группу Server Group. Присвойте ей имя Вашей студенческой группы (например, DLI-201). Выберите для нее уровень Sub-group of ... SQL Server Group. Опишите, как Вы это сделали.

4.4. Откройте окно редактирования свойств регистрации Edit SQL Server Registration properties. Присвойте регистрационную запись сервера Вашей серверной группе из 3 пункта задания. Отключите опции “Отображать состояние сервера на консоли”, “Показывать

системные базы данных и системные объекты”. Сохраните настройки. Опишите, как Вы это сделали. Что изменилось в настройках сервера?

4.5. Приостановите работу Вашего сервера, возобновите его работу, остановите Ваш сервер и запустите его снова с помощью контекстного меню Enterprise Manager. Опишите, как Вы это сделали.

4.6. Прodelайте те же действия (см. пункт 5) с помощью утилиты Service Manager. Запустите службу SQL Server Agent. Настройте автоматический запуск этой службы при запуске операционной системы. Опишите, как Вы это сделали.

4.7. Остановите сервер с помощью инструкции языка Transact-SQL SHUTDOWN. Опишите, как Вы это сделали. Для чего применяется необязательное предложение WITH NOWAIT. Каковы преимущества и недостатки применения этой опции. Запустите сервер. Как Вы это сделали?

4.8. Укажите данные об основных свойствах SQL Server:

- наименование продукта;
- операционная система;
- версия продукта;
- язык;
- платформа;
- объем оперативной памяти;
- количество процессоров;
- корневая директория сервера;
- сопоставление сервера.

Укажите, какие службы SQL Server запускаются автоматически при старте операционной системы. Какие службы не запускаются автоматически?

4.9. Установите в свойствах SQL Server метод аутентификации на основе учетных записей Windows. Настройте регистрацию в журнале VCEXпопыток подключения к серверу. Запретите другим серверам удаленно подключаться к Вашему серверу. Установите, чтобы для целостности распределенных запросов использовалась служба Distributed Transaction Coordinator. Опишите, как Вы это сделали.

4.10. Установите в свойствах SQL Server русский язык для сообщений сервера по умолчанию. Опишите, как Вы это сделали. Определите, как интерпретируются двухзначные номера (от 0 до 99) для представления года в настройках сервера.

5. Построение системы безопасности SQL SERVER

5.1. Создайте пользователя под именем sa_XXXX, где XXXX – номер Вашего студенческого билета. Установите для него аутентификацию на уровне SQLServer. Задайте для него пароль и запишите, чтобы не забыть. Установите в качестве базы данных по умолчанию Вашу базу данных (см. пункт 1). Выберите язык базы данных – русский. Укажите, что Ваш пользователь sa_XXXX является системным администратором. Запишите в отчете, какие действия может выполнять пользователь sa_XXXX. Сохраните данные о пользователе. Подтвердите пароль для пользователя. Опишите, как Вы это сделали.

5.2. Отключитесь от сервера. Отредактируйте свойства регистрационной записи Вашего сервера. Установите подключение с использованием аутентификации на уровне SQL Server. Установите - всегда запрашивать учетную запись и пароль при подключении. Подключитесь к серверу под именем Вашего пользователя sa_xxxx. Опишите, как Вы это сделали. Что означает sa в имени Вашего пользователя?

5.3. Создайте пользователя под именем User_xxxx, где xxxx – номер Вашего студенческого билета. Установите для него аутентификацию на уровне SQLServer. Задайте для него пароль и запишите, чтобы не забыть. Установите в качестве базы данных по умолчанию Northwind. Выберите язык базы данных – английский. Сохраните данные о пользователе. Подтвердите пароль для пользователя. Опишите, как Вы это сделали.

5.4. Разрешите Вашему пользователю User_xxxx доступ к базе данных Northwind. Разрешите ему считывать данные из этой базы данных. Разрешите Вашему пользователю User_xxxx доступ к базе данных Northwind_xxxx. Разрешите ему считывать и записывать данные в эту базу данных.

Разрешите Вашему пользователю User_xxxx доступ к Вашей собственной базе данных (см. пункт 1). Назначьте его собственником этой базы данных.

Запишите в отчете, какие действия может выполнять Ваш пользователь User_xxxx. Сохраните данные о пользователе. Опишите, как Вы это сделали.

5.5. Отключитесь от сервера. Подключитесь к серверу под именем Вашего пользователя User_xxxx. Откройте утилиту Query Analyzer. Сделайте текущей базу данных Northwind. Выведите информацию о служащих из таблицы Employees: Номер сотрудника, Фамилия сотрудника,, Имя сотрудника,, Год рождения сотрудника,, Возраст сотрудника, (на текущий момент времени). Измените Год рождения сотрудника, Andrew Fuller на 1982 с помощью операторов языка DML. Опишите, как Вы это сделали. Приведите текст инструкций и результаты их выполнения. Все ли операции выполнились успешно? Объясните, почему.

5.6. Сделайте текущей базу данных Northwind_xxxx. Выведите информацию о клиентах из таблицы Customers: Идентификатор клиента, Наименование компании, Город, Страна, Телефон. Измените Телефон компании Great Lakes Food Market на номер (820) 999-7667 с помощью операторов языка DML. Опишите, как Вы это сделали. Приведите текст инструкций и результаты их выполнения. Все ли операции выполнились успешно? Объясните, почему.

5.7. Создайте в базе данных Northwind_xxxx таблицу Продаж (Sales). Структура таблицы представлена ниже:

№ № п/п	Наименован ие столбца	Тип	Длина	Допустимо Null	Значен ие по умол- чанию	Ограни - чение на значен ие
1.	OrderId (№ заказа)	int		Нет		> 0

2.	CustId (№ клиента)	int		Нет		> 0
3.	SellerId (№ продавца)	int		Нет		> 0
4.	SaleDate (Дата продажи)	datetime		Нет	Текущее дата-время	
5.	SaleTotal (Сумма заказа)	Money		Да	Null	Null или >=0

Приведите команды языка DDL для создания таблицы Sales с указанными характеристиками. Опишите, как Вы это сделали. Приведите текст инструкций и результаты их выполнения. Все ли операции выполнились успешно? Объясните, почему.

5.8. Сделайте текущей Вашу собственную базу данных (см. пункт 1). Создайте в этой базе данных таблицу Продаж (Sales). Структура таблицы представлена в пункте 5.7. Приведите команды языка DDL для создания таблицы Sales с указанными характеристиками. Опишите, как Вы это сделали. Приведите текст инструкций и результаты их выполнения. Все ли операции выполнились успешно? Объясните, почему.

5.9. Отключитесь от сервера. Подключитесь к серверу под именем Вашего пользователя sa_XXXX. В Enterprise Manager выберите базу данных Northwind_XXXX, создайте в ней роль под именем MyRole. Установите тип роли – стандартная роль. Добавьте к этой роли Вашего пользователя User_XXXX. Сохраните роль. Опишите, как Вы это сделали.

5.10. Откройте для редактирования роль MyRole. Назначьте ей разрешения.

Для таблиц Categories, Products, Order Details, Suppliers, Orders, Shippers, Customers и Employees назначьте разрешения на чтение данных. Для таблиц Products, Order Details, Orders, Customers назначьте также разрешения на изменение данных. Для всех хранимых процедур назначьте разрешения на выполнение. Укажите в отчете имена этих хранимых процедур. Проверьте объекты, которые Вы назначили для этой роли, переключив режим просмотра. Сохраните изменения. Опишите, как Вы это сделали.

6. Построение резервного копирования и восстановления данных в MS SQL SERVER

6.1. Создайте резервную копию базы данных Northwind_XXXX, где XXXX – номер Вашего студенческого билета. Укажите имя резервной копии MyNorthwindBackup. Настройте полное копирование базы данных. Укажите, куда будет сохраняться база данных, укажите имя файла. Выберите “переписать существующий архив”. Опишите, как Вы это сделали.

6.2. Удалите из базы данных Northwind_XXXX с помощью команд языка DDL таблицы Categories, Products, Order Details, Suppliers, Orders, Shippers, Customers и Employees. Приведите тексты инструкций. Опишите, как Вы это сделали.

6.3. Восстановите содержимое базы данных Northwind_XXXX с помощью резервной копии MyNorthwindBackup. Опишите, как Вы это сделали. Убедитесь в наличии таблиц

Categories, Products, Order Details, Suppliers, Orders, Shippers, Customers и Employees. Укажите это в Вашем отчете.

6.4. Создайте копию базы данных Northwind_xxxx под именем New_Northwind_xxxx с помощью резервной копии MyNorthwindBackup. Опишите, как Вы это сделали.

6.5. Сделайте текущей базу данных Northwind_xxxx. Вызовите утилиту Query Analyzer. Удалите из таблицы Customers данные о клиентах с помощью команд языка DML. Приведите тексты инструкций. Опишите, как Вы это сделали.

6.6. Создайте дифференциальную резервную копию базы данных Northwind_xxxx. Укажите имя резервной копии MyNorthwindBackup. Настройте дифференциальное копирование базы данных. Укажите, куда будет сохраняться база данных, укажите имя файла.

Выберите “переписать существующий архив”. Опишите, как Вы это сделали.

6.7. Удалите базу данных New_Northwind_xxxx с помощью команд языка DDL. Приведите тексты инструкций. Опишите, как Вы это сделали.

6.8. Восстановите содержимое базы данных Northwind_xxxx из полной копии MyNorthwindBackup. Убедитесь в наличии записей о клиентах в таблице Customers. Опишите, как Вы это сделали.

6.9. Создайте копию базы данных Northwind_xxxx под именем New_Northwind_xxxx с помощью полной резервной копии MyNorthwindBackup. Затем восстановите в базу данных New_Northwind_xxxx дифференциальную копию MyNorthwindBackup. Опишите, как Вы это сделали. Укажите в отчете, есть ли в таблице Customers данные о клиентах.

6.10. Создайте план обслуживания базы данных Northwind_xxxx с помощью мастера обслуживания Database Maintenance Plan Wizard.

- Выберите, удалить неиспользованное пространство из файлов базы данных. Установите, выполнять ежемесячно в первое воскресенье каждого месяца, в 02:00:00. Выполнять до 31 декабря текущего года.

- Установите, проверять целостность базы данных.

- Укажите, создавать страховую копию на диске, проверять целостность страховой копии. Установите, выполнять ежедневно в 03:00:00. Выполнять до 28 декабря текущего года. Установите, удалять файлы старше 1 недели. Укажите расширение для файла страховой копии .BAC.

- Присвойте наименование файлу обслуживания базы данных: Maintenance Plan Northwind_xxxx.

Опишите, как Вы это сделали.

7. Использование хранимых процедур в системе безопасности

7.1. Определите имеющиеся на Вашем сервере учетные записи пользователей с помощью запроса к системной таблице syslogins базы данных master. Укажите учетные имена пользователей, базы данных по умолчанию, язык, и их фиксированные серверные роли. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите текст запроса и результат его выполнения.

7.2. С помощью системной хранимой процедуры sp_addlogin создайте новую учетную запись пользователя с именем sysadmin_xxxx, где xxxx – номер Вашего студенческого билета.

Укажите для нее пароль по Вашему усмотрению, базу данных по умолчанию - master, язык - русский. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите текст вызова этой процедуры.

7.3. С помощью системной хранимой процедуры `sp_addsrvrolemember` добавьте Вашу учетную запись `sysadmin_XXXX` к системной роли `sysadmin`.

Воспользуйтесь системой оперативной помощи Books OnLine. Укажите текст вызова этой процедуры.

7.4. Определите имеющиеся в Вашей базе `Northwind_XXXX` пользователей Windows, пользователей MS SQL Server и роли MS SQL Server с помощью запроса к системной таблице `sysusers`. Укажите имена пользователей или ролей, имеют ли они доступ к базе данных, являются ли они пользователями SQL Server, являются ли они фиксированными ролями базы данных. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите текст запроса и результат его выполнения.

7.5. С помощью системной хранимой процедуры `sp_addlogin` создайте новую учетную запись пользователя с именем `user2_XXXX`, где `XXXX` – номер Вашего студенческого билета. Укажите для нее пароль по Вашему усмотрению, базу данных по умолчанию – `Northwind_XXXX`, язык - русский.

С помощью системной хранимой процедуры `sp_grantdbaccess` предоставьте этому пользователю доступ к базе данных `Northwind_XXXX`. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите тексты вызовов обеих системных хранимых процедур.

7.6. С помощью системной хранимой процедуры `sp_addrolemember` предоставьте Вашему пользователю `user2_XXXX` возможность считывать данные из базы данных `Northwind_XXXX`. Для этого сделайте его членом фиксированной роли базы данных `db_datareader`. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите текст вызова этой процедуры.

7.7. С помощью системной хранимой процедуры `sp_addrolemember` предоставьте Вашему пользователю `user2_XXXX` возможность изменять данные в базе данных `Northwind_XXXX`. Для этого сделайте его членом фиксированной роли базы данных `db_datawriter`. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите текст вызова этой процедуры.

7.8. С помощью системной хранимой процедуры `sp_addrole` создайте новую пользовательскую роль `user_role`. Задайте для пользовательской роли `user_role` с помощью операторов GRANT разрешения на чтение записей в таблицах `Categories`, `Products`, `Order Details`, `Suppliers`, `Orders`, `Shippers`, а также разрешения на чтение, добавление, изменение и удаление записей в таблицах `Customers` и `Employees`. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите тексты вызовов процедур и инструкций.

7.9. С помощью системной хранимой процедуры `sp_addlogin` создайте новую учетную запись пользователя с именем `user3_XXXX`, где `XXXX` – номер Вашего студенческого билета. Укажите для нее пароль по Вашему усмотрению, базу данных по умолчанию – `Northwind_XXXX`, язык - русский.

С помощью оператора DENY запретите пользователю user3_xxxx чтение записей в таблице Categories. С помощью системной хранимой процедуры sp_addrolemember сделайте Вашего пользователя user3_xxxx членом пользовательской роли user_role. Воспользуйтесь системой оперативной помощи Books OnLine. Укажите тексты вызовов процедур и инструкций.

7.10. Проверьте, может ли пользователь user3_xxxx считывать записи из таблиц Categories и Products. Проверьте, может ли он считывать, добавлять, изменять и удалять записи в таблицах Customers и Employees. Укажите, как Вы это сделали. Какие получились результаты?

Отмените разрешение роли user_role на чтение записей в таблице Products с помощью оператора REVOKE. Что изменилось для пользователя user3_xxxx? Как Вы это определили?

Теоретический и практический материал

1. Создание БД в СУБД Microsoft Access

MS Access является частью пакета MS Office и представляет собой СУБД. Запустите MS Access, используя команды меню «Пуск» -> «Программы» -> «Microsoft Office» -> «Access». Перед вами откроется основное окно программы. Выполните пункты меню «Файл» - «Создать», после чего из вариантов выберите «Новая база данных» и сохраните вашу БД на диск.

Открывшееся окно представляет собой новую БД.

Объекты БД в среде MS Access

MS Access поддерживает следующие объекты:

- Таблица – отношение в реляционной терминологии;
- Запрос – сохраненный текст (модель) запроса на языке SQL к таблицам БД;
- Формы – конструирование интерфейса пользователя;
- Отчеты – конструирование выходной информации БД;
- Страницы – страницы доступа через web;
- Макросы – группы макрокоманд;
- Модули – программные модули на языке VBA, которые могут быть использованы в запросах, формах, отчетах, макросах.

Создание таблиц в среде MS Access

СУБД MS Access поддерживает различные варианты создания таблиц. Рассмотрим наиболее часто используемый вариант создания – в режиме конструктора. Для этого среди объектов БД выберите «Таблицы», после чего выполните пункт «Создание таблицы в режиме конструктора».

В появившемся окне предлагается ввести имена атрибутов таблицы с указанием их типов. Типы атрибутов выбираются из выпадающего списка вариантов, а конкретные характеристики указываются на панели внизу (например, для текстового формата можно указать максимальный размер). Третий столбец на форме конструирования отношений – описание атрибутов. При создании сложных БД с большим числом отношений и атрибутов в них обязательно описывайте логику, которую вы закладываете в тот или иной атрибут.

После создания всех атрибутов не забудьте указать ключевое поле вашего отношения. Для этого выделите нужные поля и нажмите на иконку «ключевое поле» изображающую ключ на панели конструктора.

Создание схем данных в среде MS Access

СУБД MS Access реализует инструмент проектирования связей между таблицами (объектами) БД. Этот инструмент называется «Схема данных».

Для вызова этого инструмента можете воспользоваться соответствующей кнопкой на панели «База данных» (рис. 1.1)



Рисунок 1.1 – Панель «База данных»

В рабочую область схемы можно добавлять и удалять таблицы, а также связывать между ними. Для добавления связи между таблицами выберите необходимое поле (внешний ключ) и используя Drag'n'Drop протяните его до нужного поля связанной таблицы (потенциальный ключ). После чего, в появившемся окне свойств вновь созданной связи, можете выставить необходимые настройки (рис.1.2).

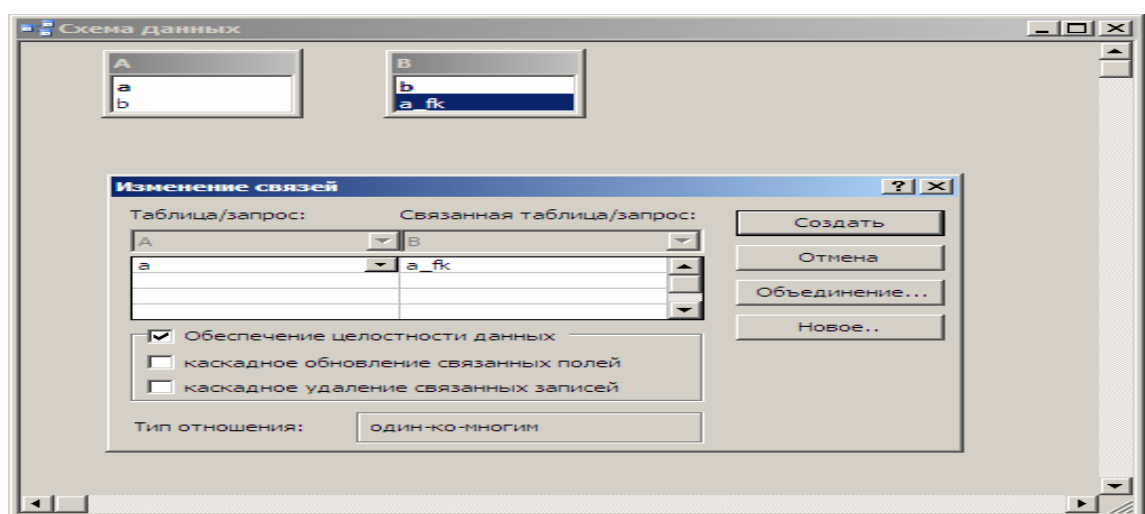


Рисунок 1.2 Редактирование связей в MS Access После окончания редактирования не забудьте сохранить схему.

2. СОЗДАНИЕ БАЗЫ ДАННЫХ В СРЕДЕ MICROSOFT SQL SERVER

Таблица 2.1. Соответствие типов данных Microsoft Access и Microsoft SQL

№	Тип данных Microsoft Access	Тип данных Microsoft SQL	Описание типа данных Microsoft SQL
1	Текстовый	nvarchar	Тип данных для хранения текста до 4000 символов
2	Поле MEMO	ntext	Тип данных для хранения символов в кодировке Unicode до 1 073 741 823 символов
3	Числовой	int	Численные значения (целые) в диапазоне от -2 147 483 648 до +2 147 483 647
4	Дата/время	smalldatetime	Дата и время от 1 января 1900 г. до 6 июня 2079 года с точностью до одной минуты

5	Денежный	money	Денежный тип данных, значения которого лежат в диапазоне от -922 337 203 685 477.5808 до +922 337 203 685 477.5807, с точностью до одной десятичной
6	Счетчик	int	См. пункт 3
7	Логический	bit	Переменная, способная принимать только два значения - 0 или 1
8	Поле объекта OLE	image	Переменная для хранения массива байтов от 0 до 2 147 483 647 байт
9	Гиперссылка	ntext	См. пункт 2
10	Мастер подстановок	nvarchar	См. пункт 1

Пример создания базы данных Microsoft SQL Server в MS Visual Studio

В данной теме показано решение задачи создания базы данных типа SQL Server с помощью MS Visual Studio. Рассматриваются следующие вопросы:

- работа с окном Server Explorer в MS Visual Studio;
- создание локальной базы данных типа SQL Server Database;
- создание таблиц в базе данных;
- редактирование структур таблиц;
- связывание таблиц базы данных между собой;
- внесение данных в таблицы средствами MS Visual Studio.

Условие задачи

Используя средства [MS Visual Studio](#) создать базу данных типа [MS SQL Server](#) с именем [Education](#). База данных содержит две таблицы [Student](#) и [Session](#). Таблицы между собой связаны по некоторому полю.

Структура первой таблицы «[Student](#)».

Название поля	Тип поля	Объяснение
<u>Num_book</u>	Текстовый	Номер зачетной книжки
<u>Name</u>	Текстовый	Фамилия, имя, отчество студента
<u>Group</u>	Текстовый	Название группы, в которой учится студент
<u>Year</u>	Целое число	Год рождения

Структура второй таблицы «[Session](#)».

Название поля	Тип поля	Объяснение
<u>Num_book</u>	Текстовый	Номер зачетной книжки
<u>Mathematics</u>	Целое число	Оценка по математике
<u>Informatics</u>	Целое число	Оценка из информатики
<u>Philosophy</u>	Целое число	Оценка из философии

Выполнение

1. Загрузить **MS Visual Studio**.

2. Активировать окно **Server Explorer**. (обозреватель серверов)

Для работы с базами данных корпорация **Microsoft** предлагает облегченный сервер баз данных **Microsoft SQL Server**.

Прежде всего, перед созданием базы данных, нужно активировать утилиту **Server Explorer**. Для этого, в **MS Visual Studio** нужно вызвать (рис. 2.1)

View -> Server Explorer

(в некоторых версиях – обозреватель серверов находится в вид-другие объекты)

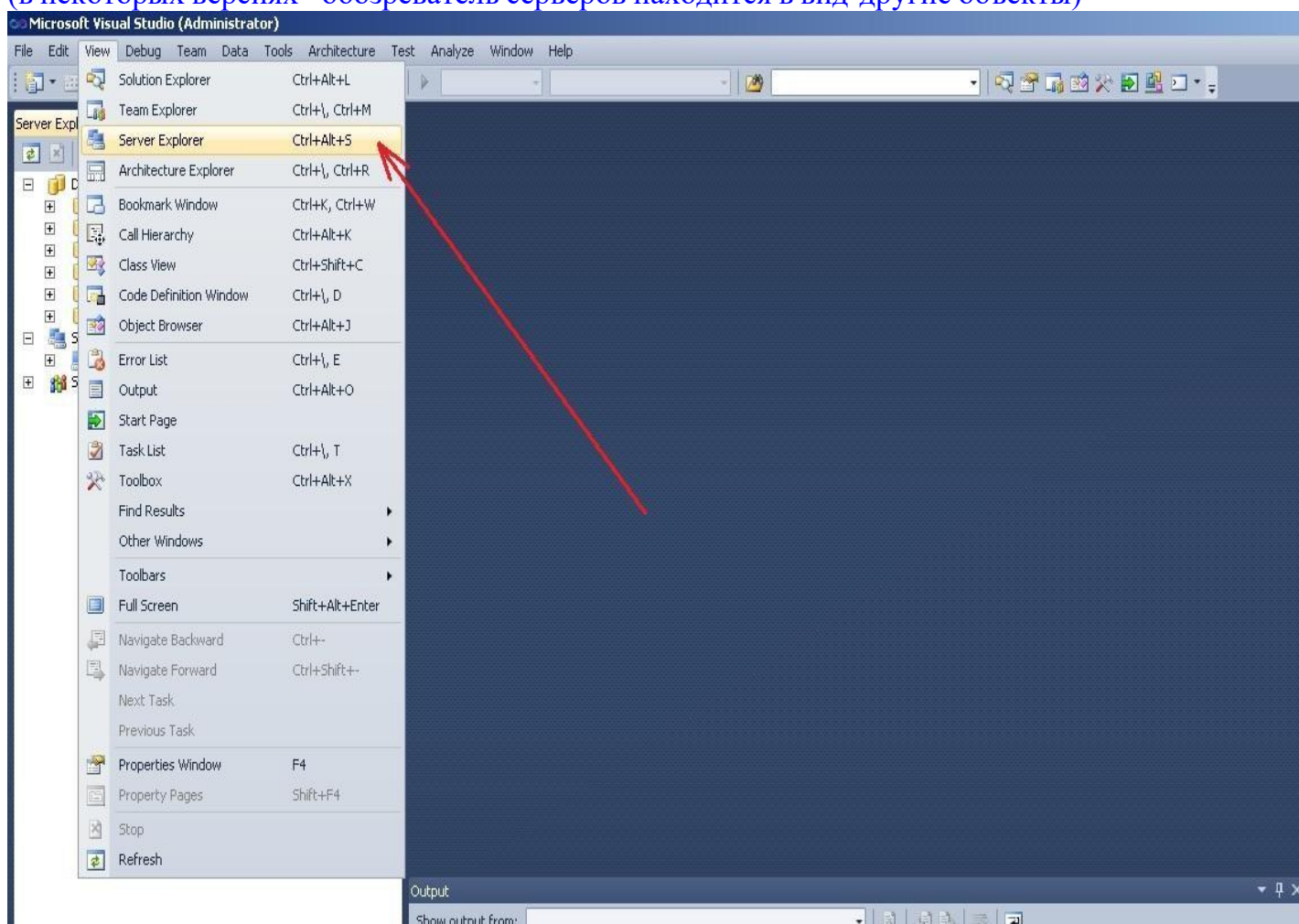


Рис. 2.1. Вызов **Server Explorer**

После вызова окно **Server Explorer** будет иметь приблизительный вид, как показанона

рисунке 2.2.

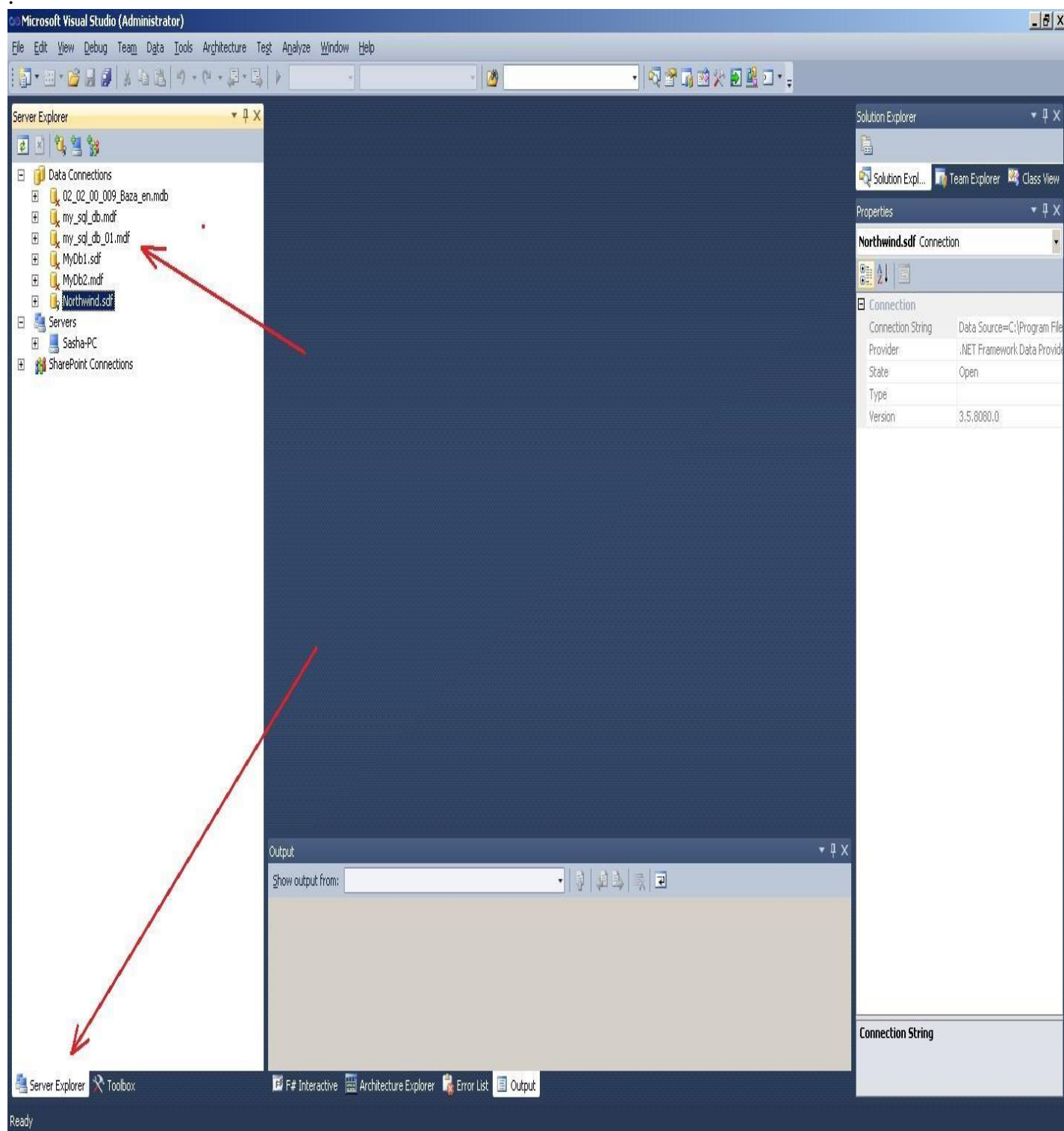


Рис. 2.2. Окно **Server Explorer**

3. Создание базы данных **“Education”**.

Чтобы создать новую базу данных, базирующуюся на поставщике данных **MicrosoftSQL Server**, нужно кликнуть на узле **Data Connections**, а потом выбрать **“Create New SQL Server Database ...”** (рис. 2.3).

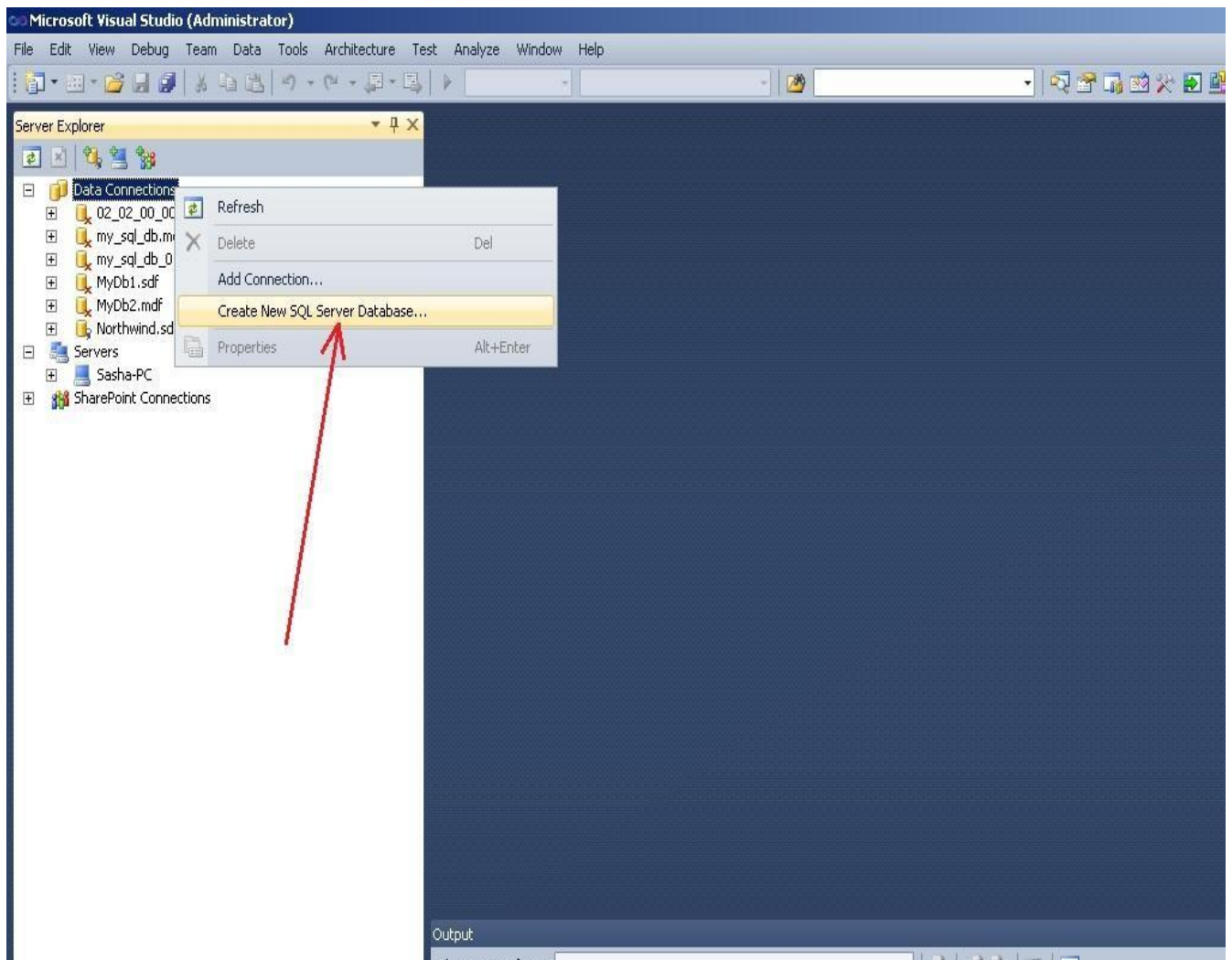


Рис. 2.3. Вызов команды создания базы данных **SQL Server**

В результате откроется окно «**Create New SQL Server Database**» (рис. 2.4).

В окне (в поле «**Server Name**») указывается имя локального сервера, установленного на вашем компьютере. В нашем случае это имя «**SQLEXPRESS**».

В поле «**New database name:**» указывается имя создаваемой базы данных. В нашем случае это имя **Education**.

Опцию **Use Windows Autentification** нужно оставить без изменений и нажать кнопку **OK**.



Рис. 2.4. Создание новой базы данных [SQL Server](#) с помощью [MS Visual Studio](#)

После выполненных действий, окно [Server Explorer](#) примет вид, как показано на рисунке 2.5. Как видно из рисунка 2.5, в список имеющихся баз данных добавлена база данных [Education](#) с именем [sasha-pc\sqlexpress.Education.dbo](#)

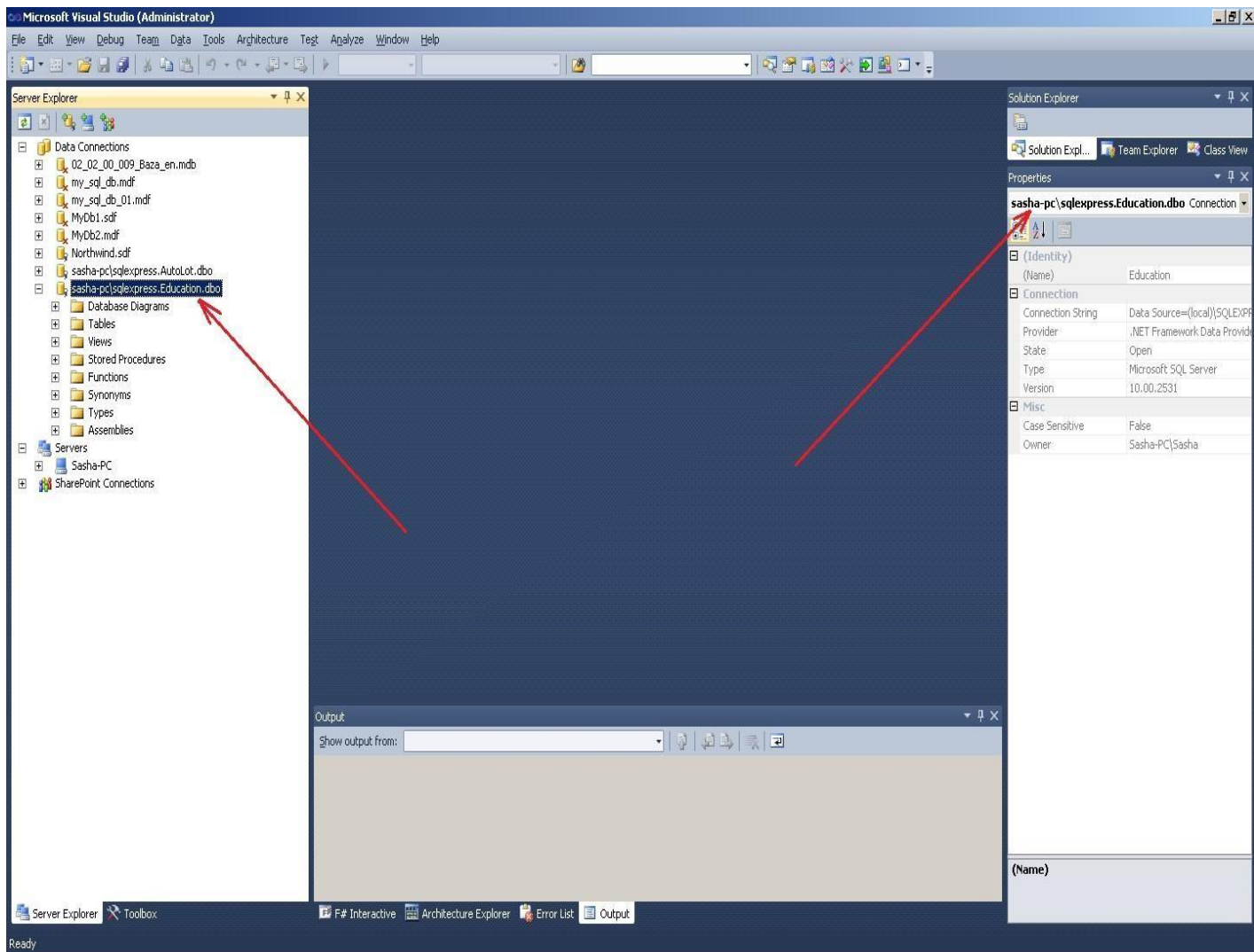


Рис. 2.5. Окно **Server Explorer** после добавления базы данных **Education**

4. Объекты базы данных **Education**.

Если развернуть базу данных **Education** (знак «+»), то можно увидеть список из следующих основных объектов:

- **Database Diagrams** – диаграммы базы данных. Диаграммы показывают связи между таблицами базы данных, отношения между полями разных таблиц и т.п.;
- **Tables** – таблицы, в которых помещаются данные базы данных;
- **Views** – представления. Отличие между представлениями и таблицами состоит в том, что таблицы баз данных содержат данные, а представления данных не содержат их, а содержимое выбирается из других таблиц или представлений;
- **Stored procedures** – хранимые процедуры. Они представляют собою группу связанных операторов на языке **SQL**, что обеспечивает дополнительную гибкость при работе с базой данных.

5. Создание таблицы **Student**.

На данный момент база данных **Education** абсолютно пустая и не содержит никаких объектов (таблиц, сохраненных процедур, представлений и т.д.).

Чтобы создать таблицу, нужно вызвать контекстное меню (клик правой кнопкой мышки) и выбрать команду “**Add New Table**” (рисунок 2.6).

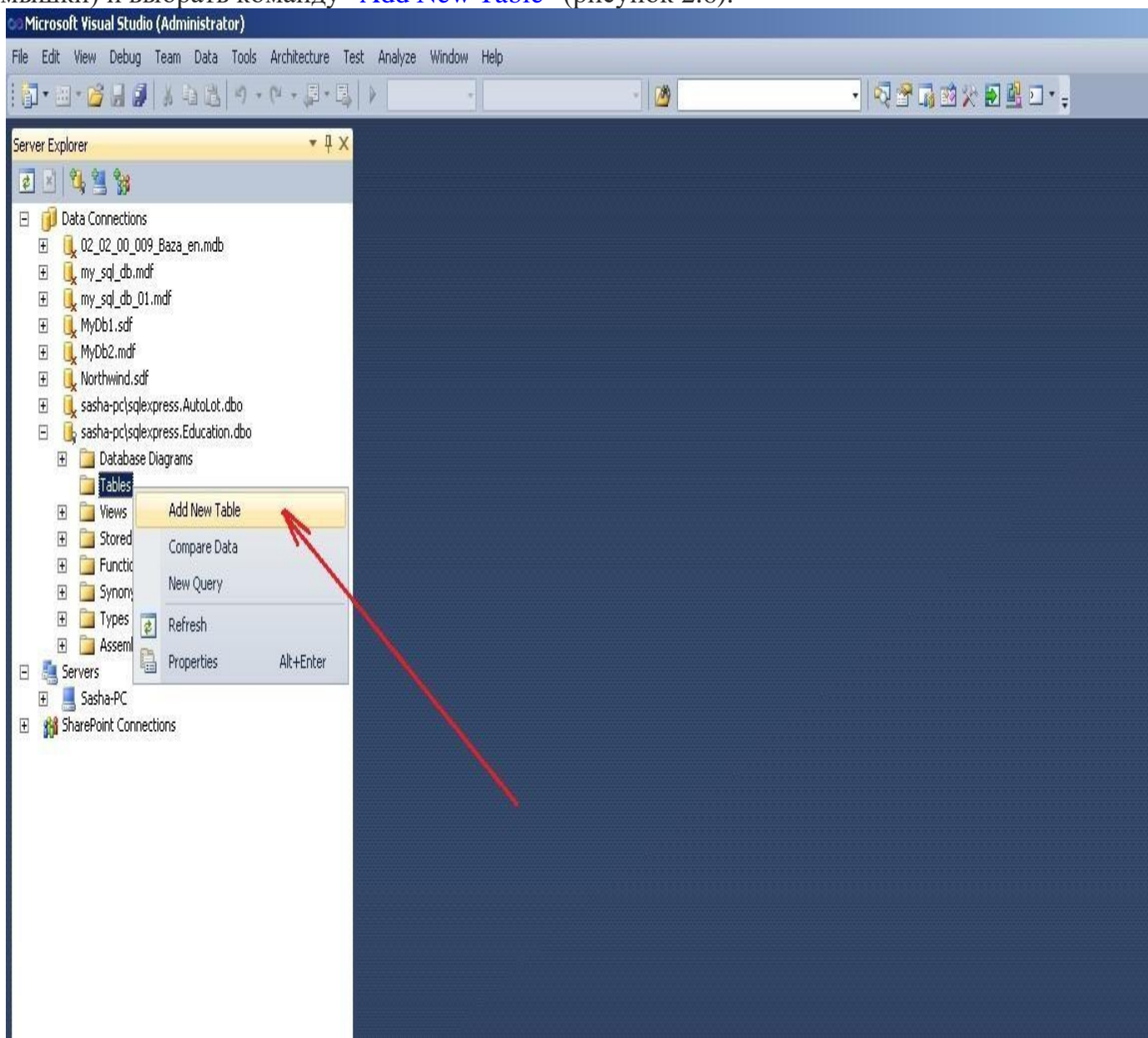


Рис. 2.6. Команда добавления новой таблицы

Существует и другой вариант добавления таблицы базы данных с помощью команд

меню Data:

Data -> Add New -> Table

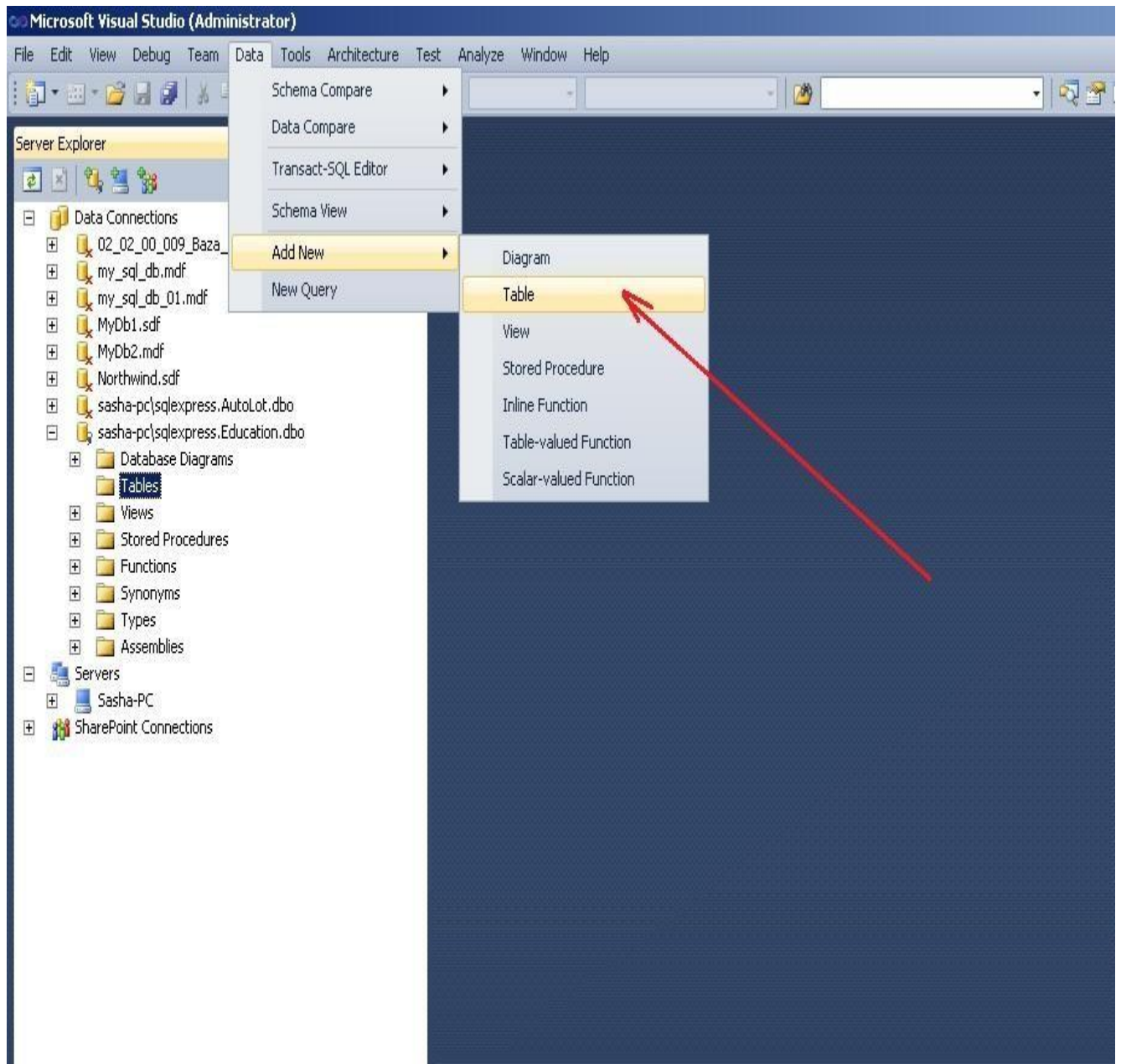


Рис. 2.7. Альтернативный вариант добавления новой таблицы

В результате откроется окно добавления таблицы, которое содержит три столбца (рисунок 2.8). В первом столбце “Column Name” нужно ввести название соответствующего поля таблицы базы данных. Во втором столбце “Data Type” нужно ввести тип данных этого поля. В третьем столбце “Allow Nulls” указывается опция о возможности отсутствия данных в поле.

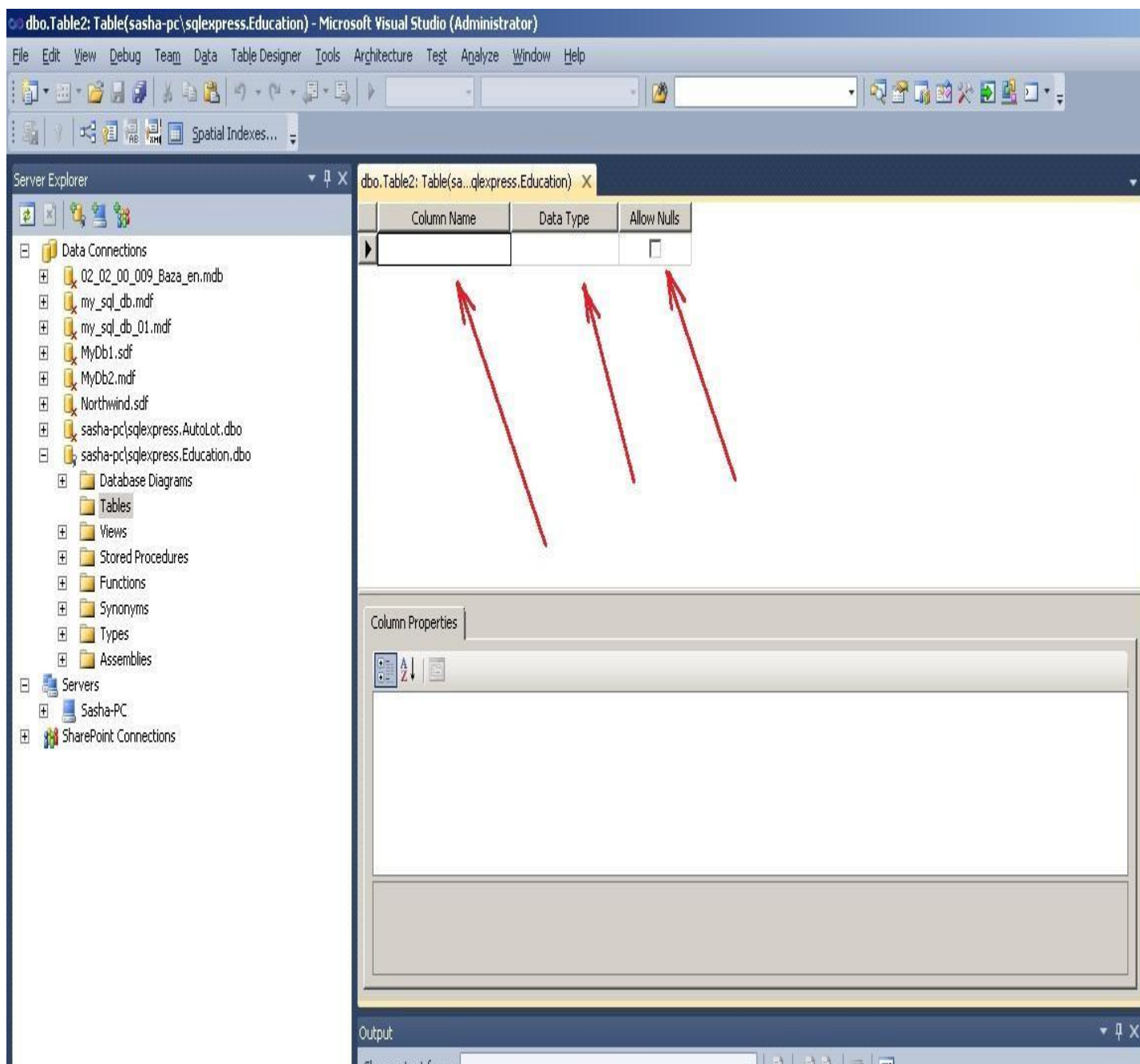


Рис. 2.8. Окно создания новой таблицы

С помощью редактора таблиц нужно сформировать таблицу [Student](#) как изображена на рисунке 2.9. Имя таблицы нужно задать при ее закрытии.

В редакторе таблиц можно задавать свойства полей в окне [Column Properties](#). Для того, чтобы задать длину строки ([nvarchar](#)) в символах, в окне [Column Properties](#) есть свойство [Length](#). По умолчанию значения этого свойства равно 10.

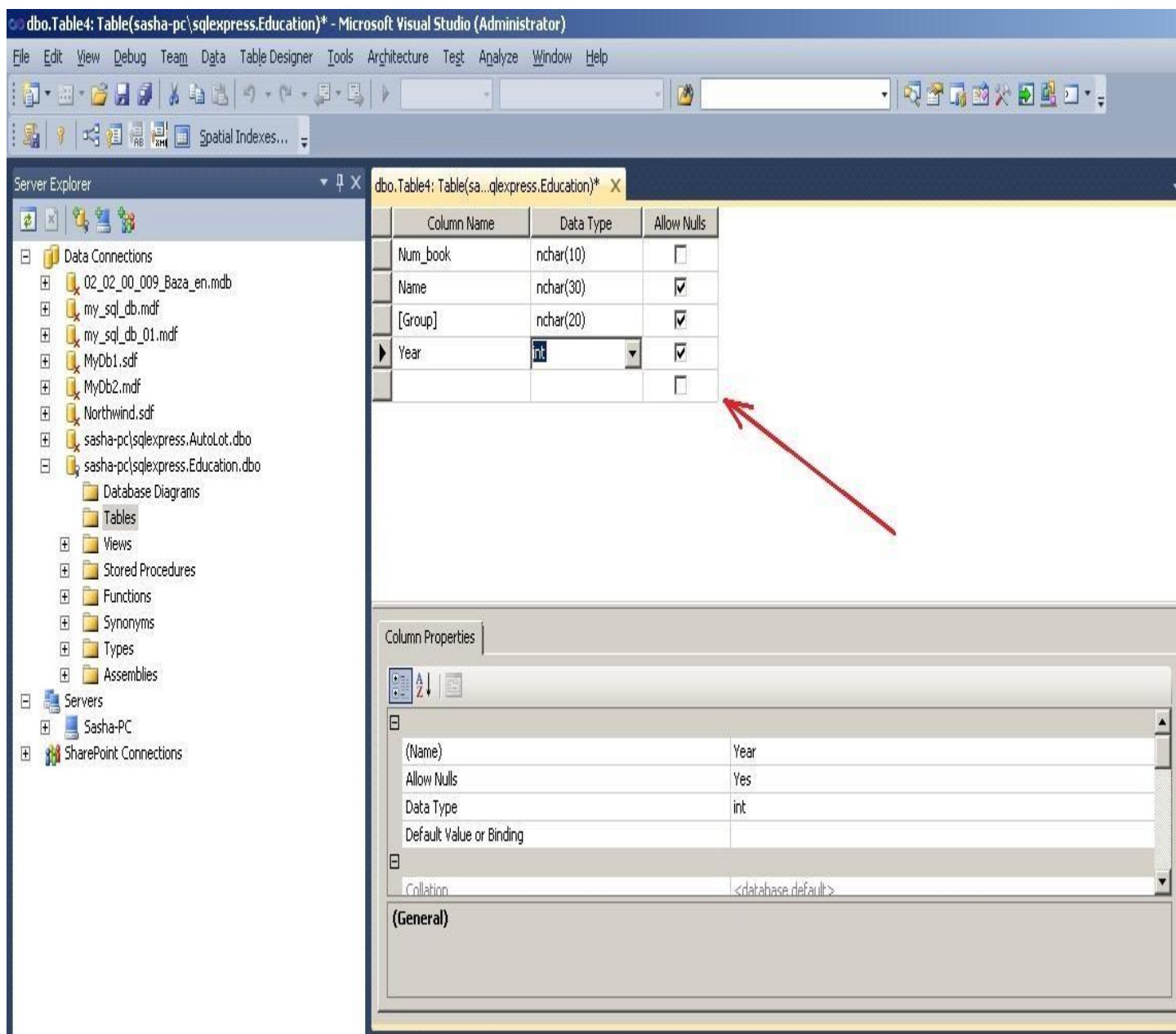


Рис. 2.9. Таблица **Student**

Следующим шагом нужно задать ключевое поле. Это осуществляется вызовом команды “**Set Primary Key**” из контекстного меню поля **Num_book**. С помощью ключевого поля будут установлены связи между таблицами. В нашем случае ключевым полем есть номер зачетной книжки.

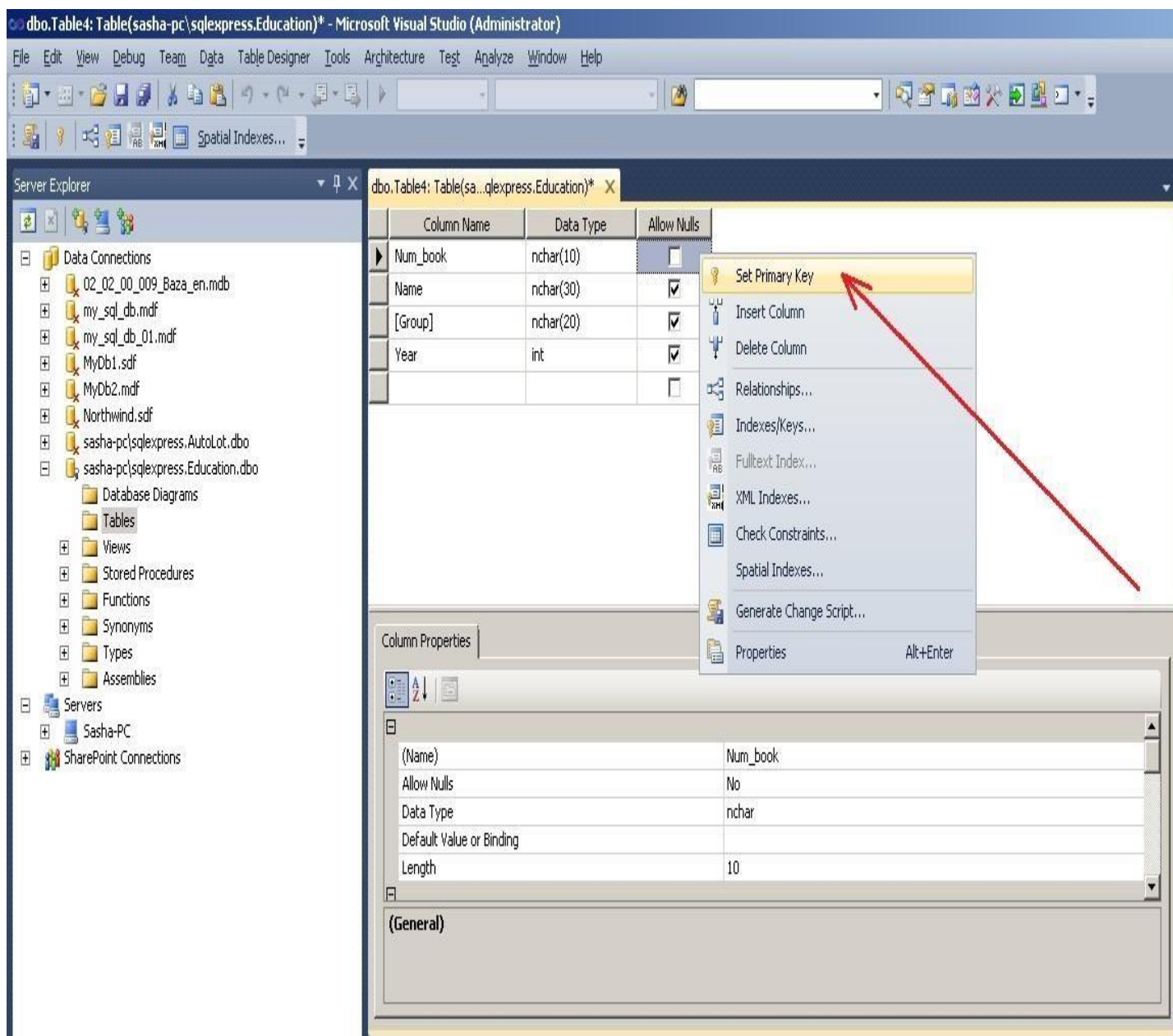


Рис. 2.10. Задание ключевого поля

После установки первичного ключа окно таблицы будет иметь вид как изображена на рисунке 2.11.

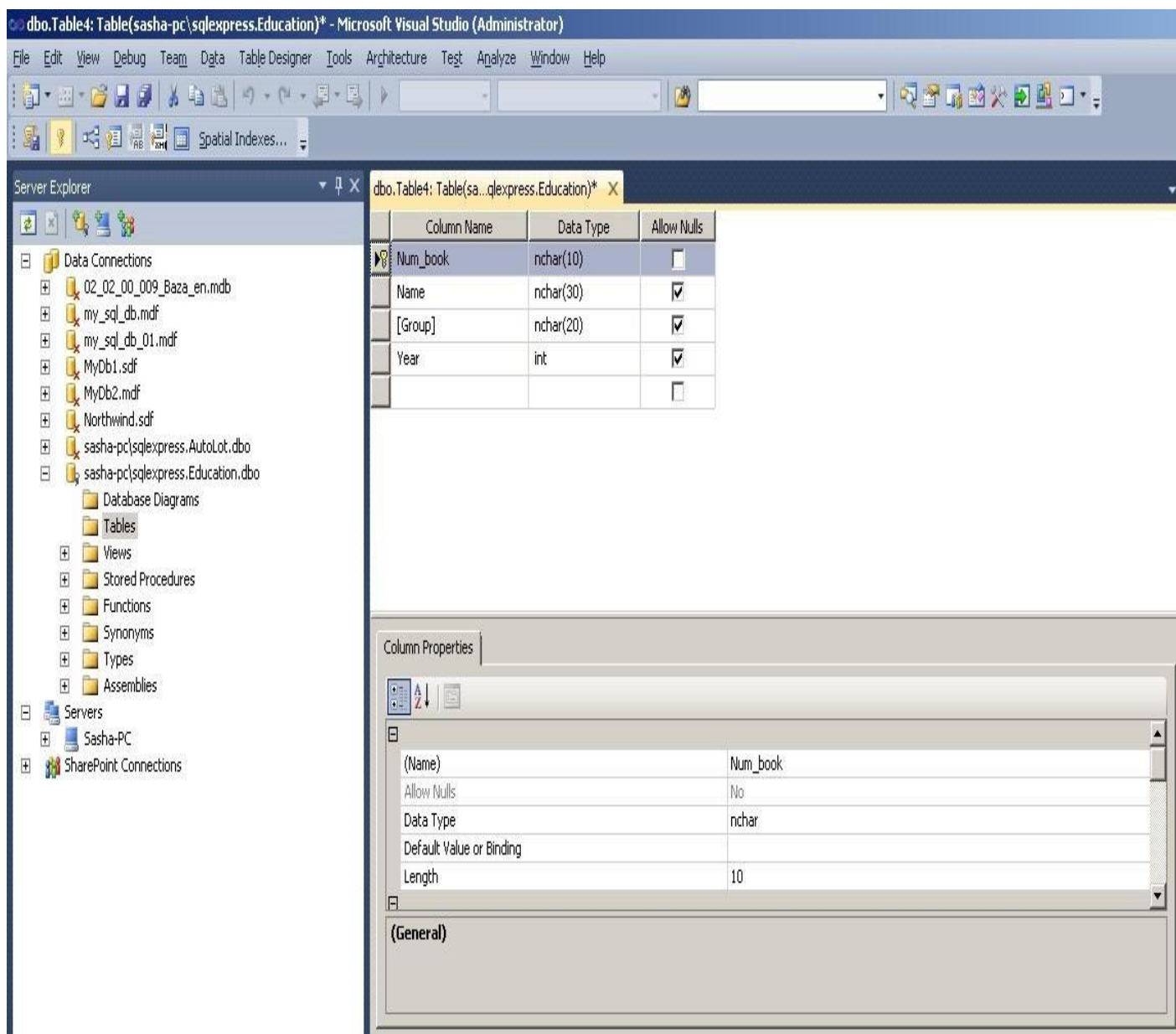


Рис. 2.11. Таблица **Student** после окончательного формирования
Теперь можно закрыть таблицу. В окне сохранения таблицы нужно задать ее имя – **Student** (рис. 2.12).



Рис. 2.12. Ввод имени таблицы **Student**

6. Создание таблицы **Session**.

По образцу создания таблицы **Student** создается таблица **Session**.

На рисунке 2.13 изображен вид таблицы **Session** после окончательного формирования. Первичный ключ (**Primary Key**) устанавливается в поле **Num_book**. Имя таблицы задается **Session**.

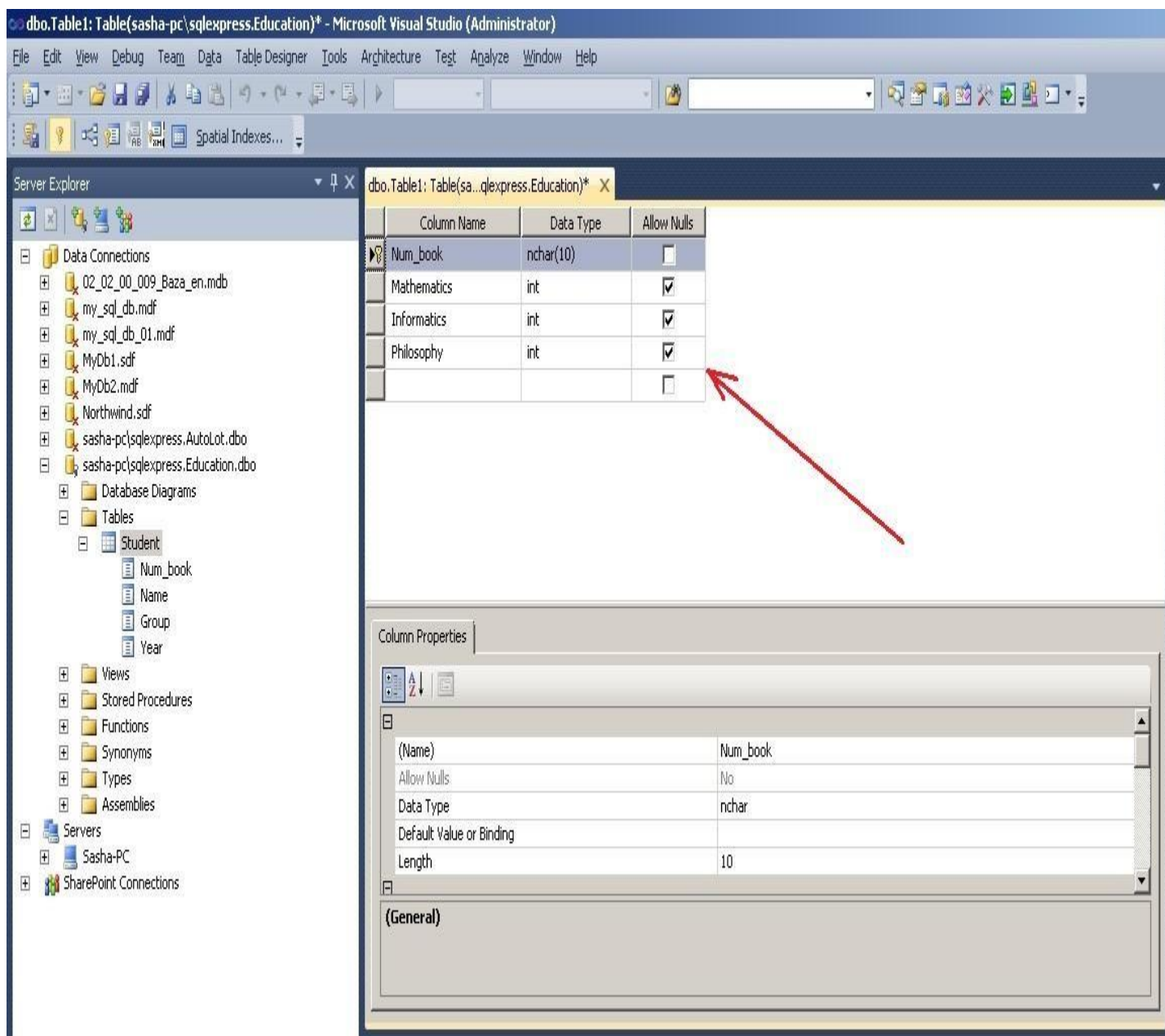


Рис. 2.13. Таблица **Session**

После выполненных действий, в окне **Server Explorer** будут отображаться две таблицы **Student** и **Session**.

Таким образом, в базу данных можно добавлять любое количество таблиц.

7. Редактирование структуры таблиц.

Бывают случаи, когда нужно изменить структуру таблицы базы данных.

Для того, чтобы вносить изменения в таблицы базы данных в **MS VisualStudio**, сначала нужно снять опцию **“Prevent Saving changes that require table re-creation”** как показано на рисунке 2.14. Иначе, **MS Visual Studio** будет блокировать внесения изменений в ранее созданную таблицу. Окно **Options**, показанное на рисунке 14 вызывается из меню **Tools** в такой последовательности:

Tools -> Options -> Database Tools -> Table and Database Designers

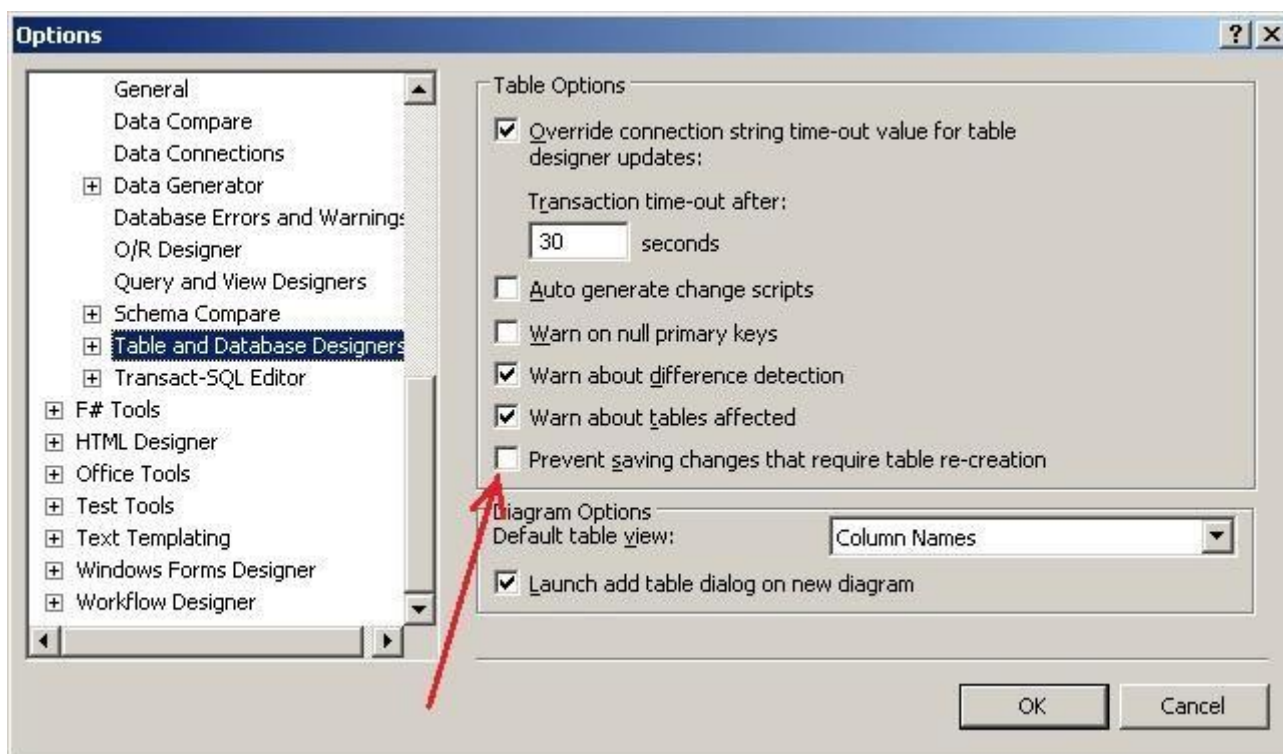


Рис. 2.14. Опция “[Prevent Saving changes that require table re-creation](#)”

После настройки можно изменять структуру таблицы. Для этого используется команда “[Open Table Definition](#)” (рисунок 2.15) из контекстного меню, которая вызывается для выбранной таблицы (правый клик мышкой).

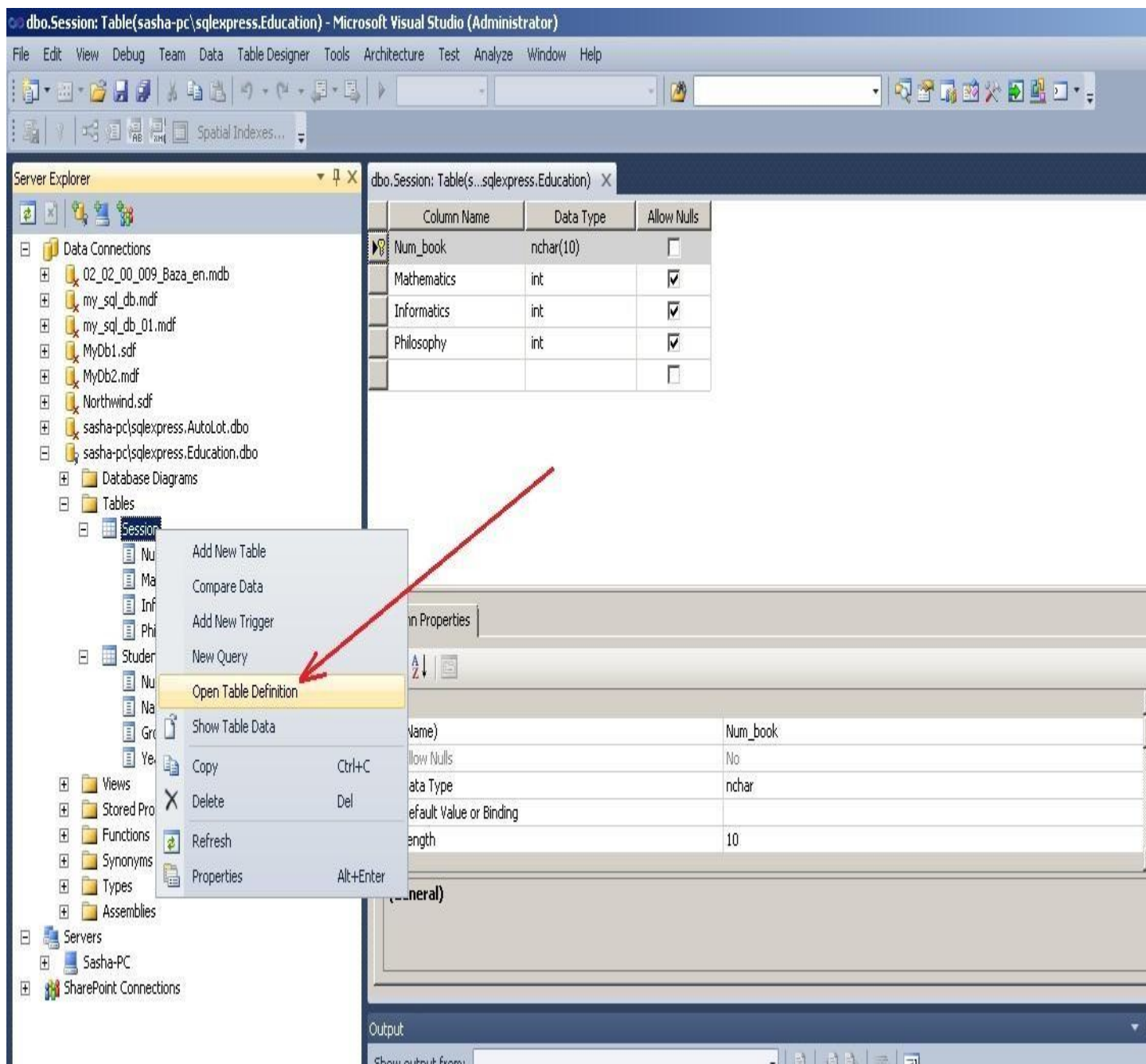


Рис. 2.15. Вызов команды “Open Table Definition” Также эта команда

размещается в меню **Data:**

Data -> Open Table Definition

Предварительно таблицу нужно выделить.

8. Установление связей между таблицами.

В соответствии с условием задачи, таблицы связаны между собою по полю **Num_book**.

Чтобы создать связь между таблицами, сначала нужно (рисунок 16):

- выделить объект [Database Diagram](#);
- выбрать команду [Add New Diagram](#) из контекстного меню (или из меню [Data](#)).

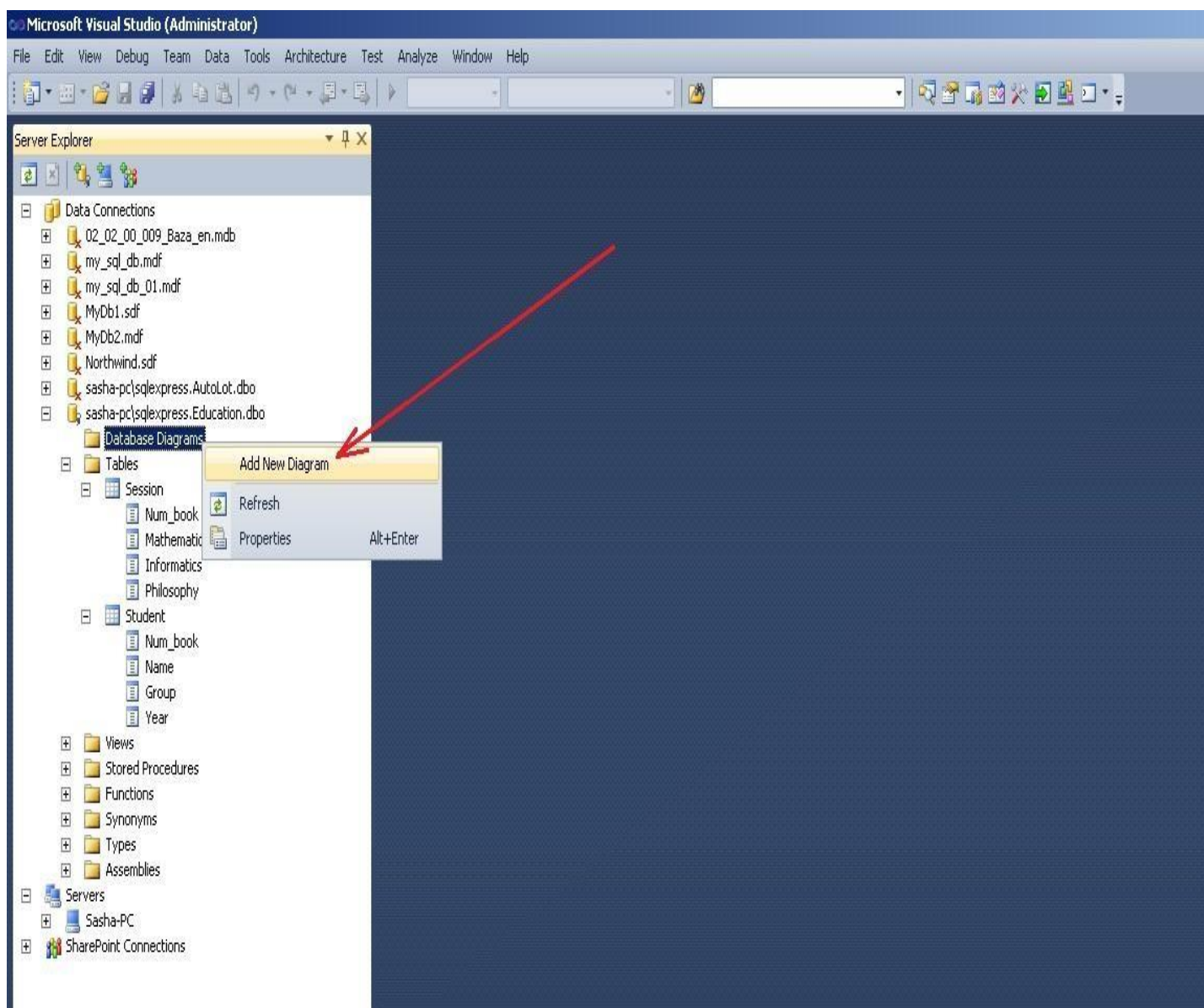


Рис. 2.16. Вызов команды добавления новой диаграммы

В результате откроется окно добавления новой диаграммы [Add Table](#) (рисунок 2.17). В этом окне нужно выбрать последовательно две таблицы [Session](#) и [Student](#) и нажать кнопку [Add](#).

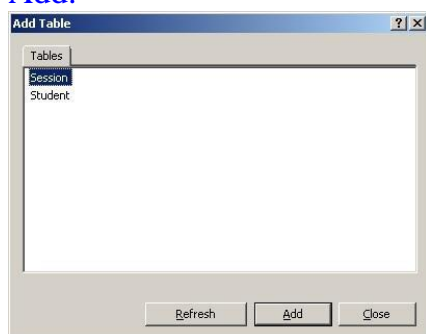


Рис. 2.17. Окно добавления таблиц к диаграмме

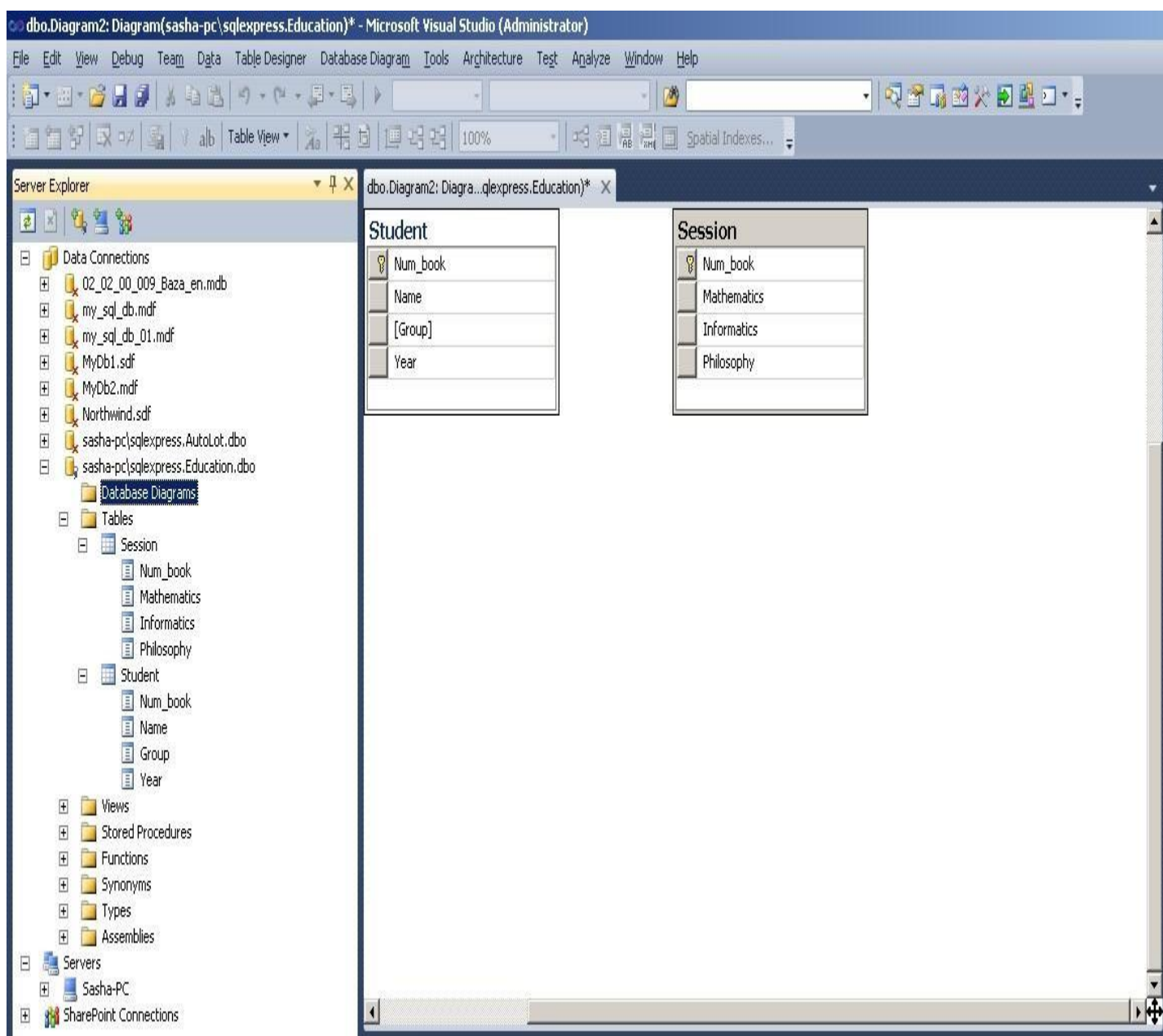


Рис. 2.18. Таблицы **Student** и **Session** после добавления их к диаграмме

Чтобы начать устанавливать отношение между таблицами, надо сделать клик на поле **Num_book** таблицы **Student**, а потом (не отпуская кнопку мышки) перетянуть его на поле **Num_book** таблицы **Session**.

В результате последовательно откроются два окна: **Tables and Columns** (рис. 2.19) и **Foreign Key Relationship** (рис. 2.20), в которых нужно оставить все как есть и подтвердить свой выбор на **OK**.

В окне **Tables and Columns** задается название отношения (**FK_Session_Student**) и названия родительской (**Student**) и дочерней таблиц.

Tables and Columns

Relationship name:

Primary key table: Foreign key table:

Primary key table	Foreign key table
Num_book	Num_book

OK Cancel

Рис. 2.19. Окно **Tables and Columns**

Foreign Key Relationship

Selected Relationship:

Editing properties for new relationship. The 'Tables And Columns Specification' property needs to be filled in before the new relationship will be accepted.

<input type="checkbox"/>	
<input type="checkbox"/>	Check Existing Data On Create Yes
<input checked="" type="checkbox"/>	Tables And Columns Specification
<input type="checkbox"/>	
<input type="checkbox"/>	Enforce For Replication Yes
<input type="checkbox"/>	Enforce Foreign Key Constraint Yes
<input checked="" type="checkbox"/>	INSERT And UPDATE Specific
<input type="checkbox"/>	
(Name)	FK_Session_Student
Description	

OK Cancel

Рис. 2.20. Окно настройки свойств отношения

После выполненных действий будет установлено отношение между таблицами (рисунок 2.21).

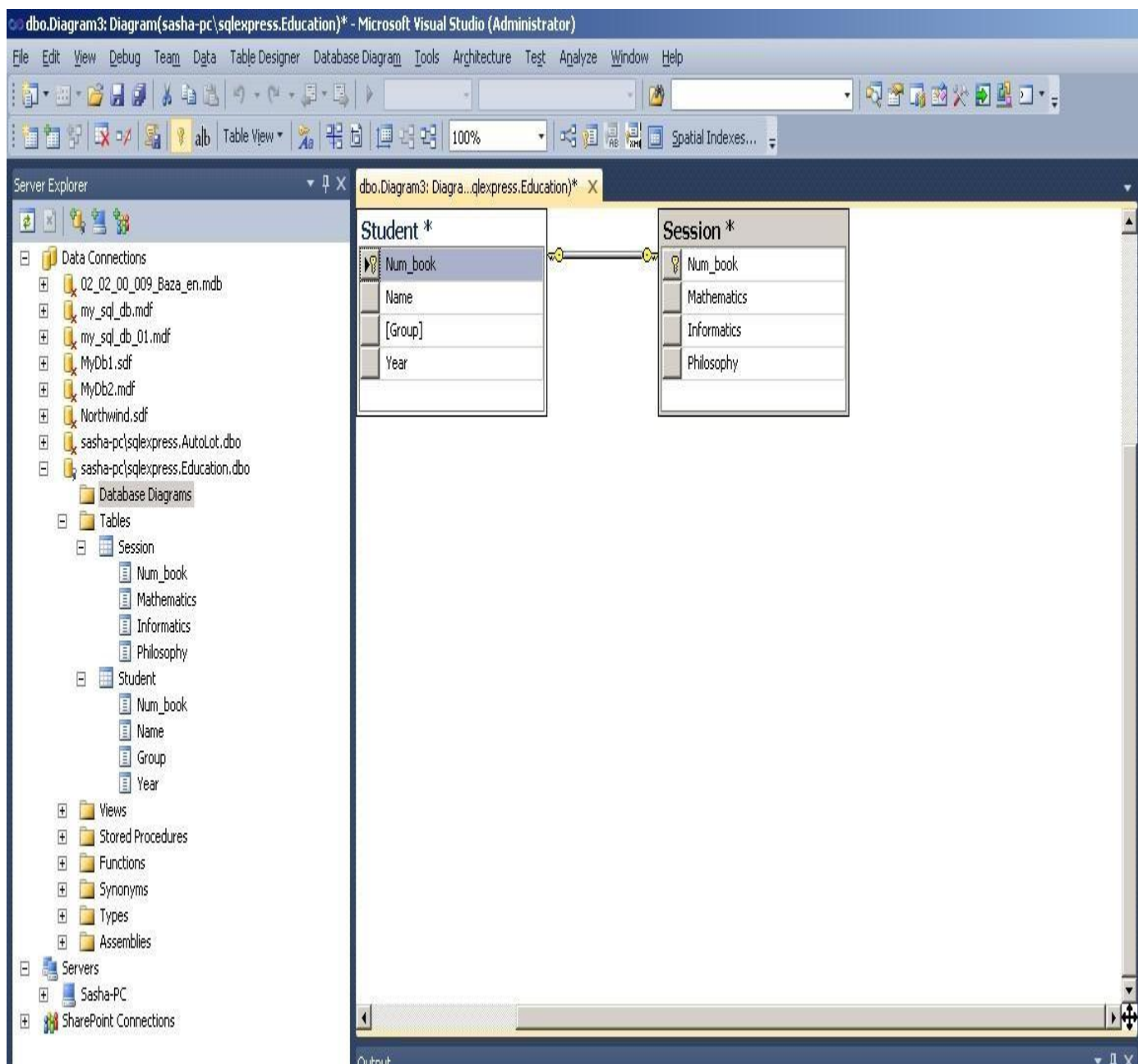


Рис. 2.21. Отношение между таблицами **Student** и **Session**

Сохранение диаграммы осуществляется точно также как и сохранение таблицы. Имя диаграммы нужно выбрать на свое усмотрение (например **Diagram1**). После задания имени диаграммы откроется окно **Save**, в котором нужно подтвердить свой выбор (рисунок 2.22).



Рис. 2.22. Подтверждение сохранения изменений в таблицах

9. Внесение данных в таблицы.

Система [Microsoft Visual Studio](#) позволяет непосредственно вносить данные в таблицы базы данных.

В нашем случае, при установлении связи (рис. 19) первичной ([Primary Key Table](#)) избрана таблица [Student](#). Поэтому, сначала нужно вносить данные в ячейки именно этой таблицы. Если попробовать сначала внести данные в таблицу [Session](#), то система заблокирует такой ввод с выводом соответствующего сообщения.

Чтобы вызвать режим ввода данных в таблицу [Student](#), нужно вызвать команду [Show Table Data](#) из контекстного меню (клик правой кнопкой мышки) или сменю [Data](#) (рис. 2.23).

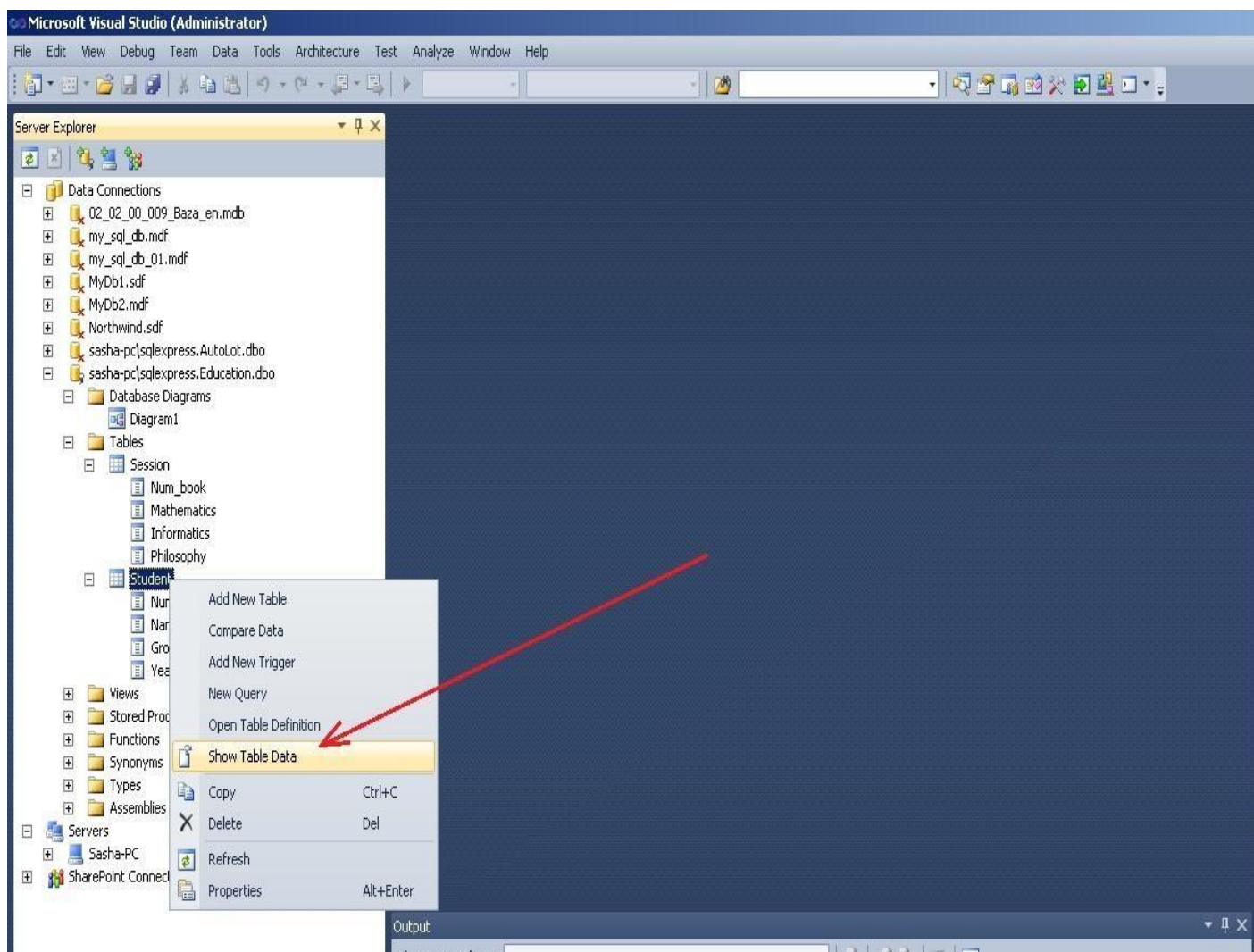


Рис. 2.23. Команда **Show Table Data**

Откроется окно, в котором нужно ввести входные данные (рис. 2.24).

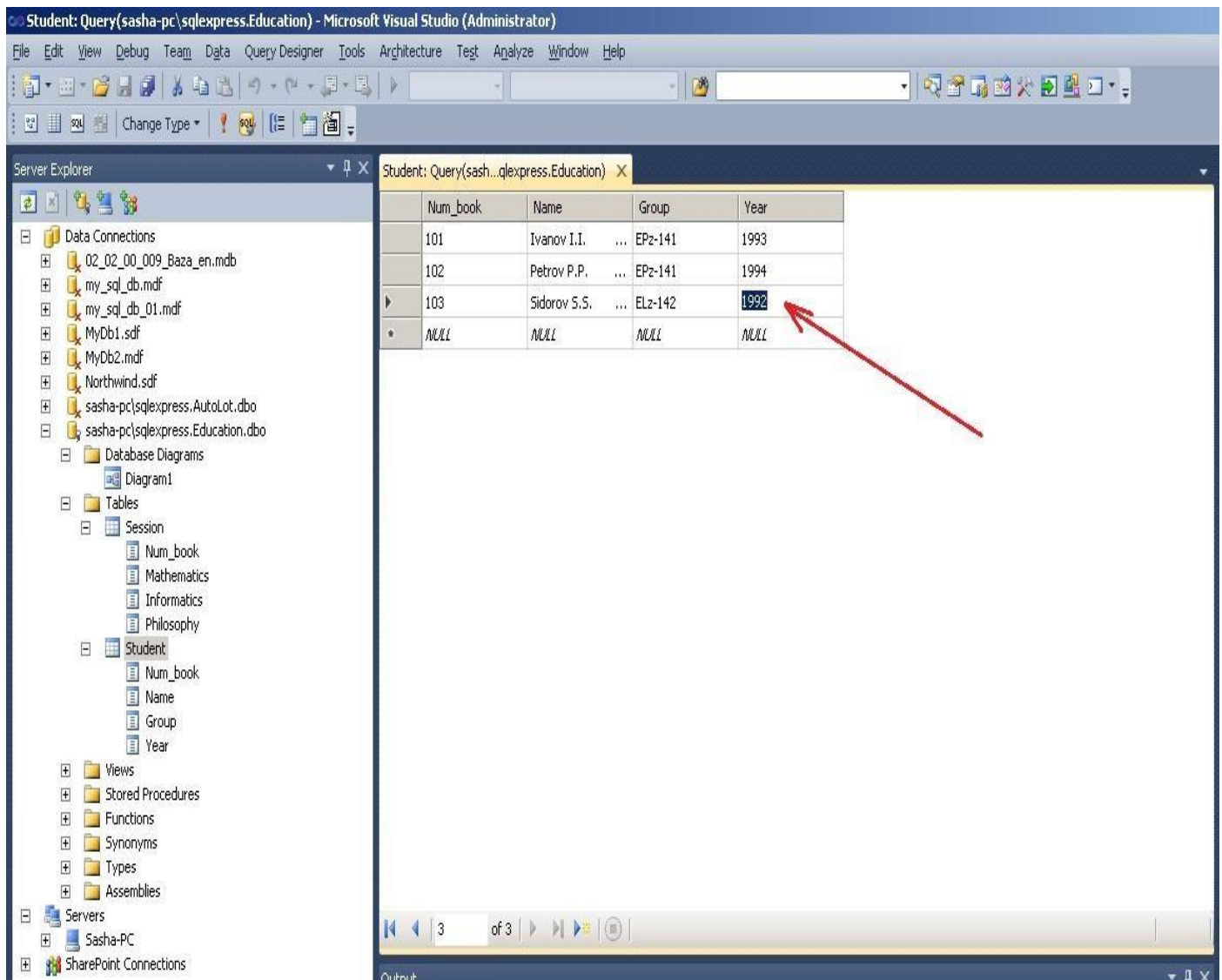


Рис. 2.24. Ввод данных в таблице **Student**

После внесения данных в таблицу **Student** нужно внести данные в таблицу **Session**.

При внесении данных в поле **Num_book** таблицы **Session** нужно вводить точно такие же значения, которые введены в поле **Num_book** таблицы **Student** (поскольку эти поля связаны между собой).

Например, если в поле **Num_book** таблицы **Student** введены значения “101”, “102”, “103” (см. рис. 2.24), то следует вводить именно эти значения в поле **Num_book** таблицы **Session**. Если попробовать ввести другое значение, система выдаст приблизительно следующее окно (рис. 2.25).

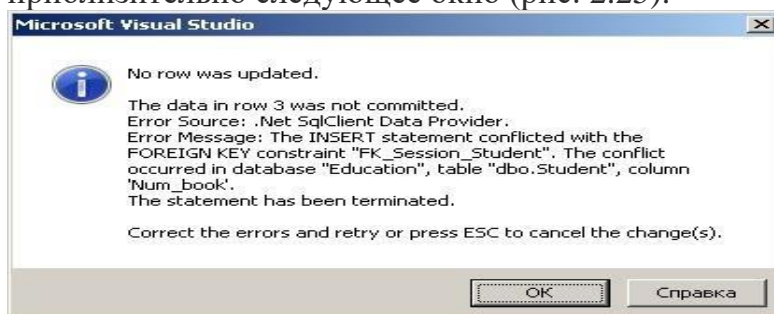


Рис. 2.25. Сообщение об ошибке ввода данных связанных таблиц **Student** и **Session**
Таблица **Session** с введенными данными изображена на рисунке 2.26.

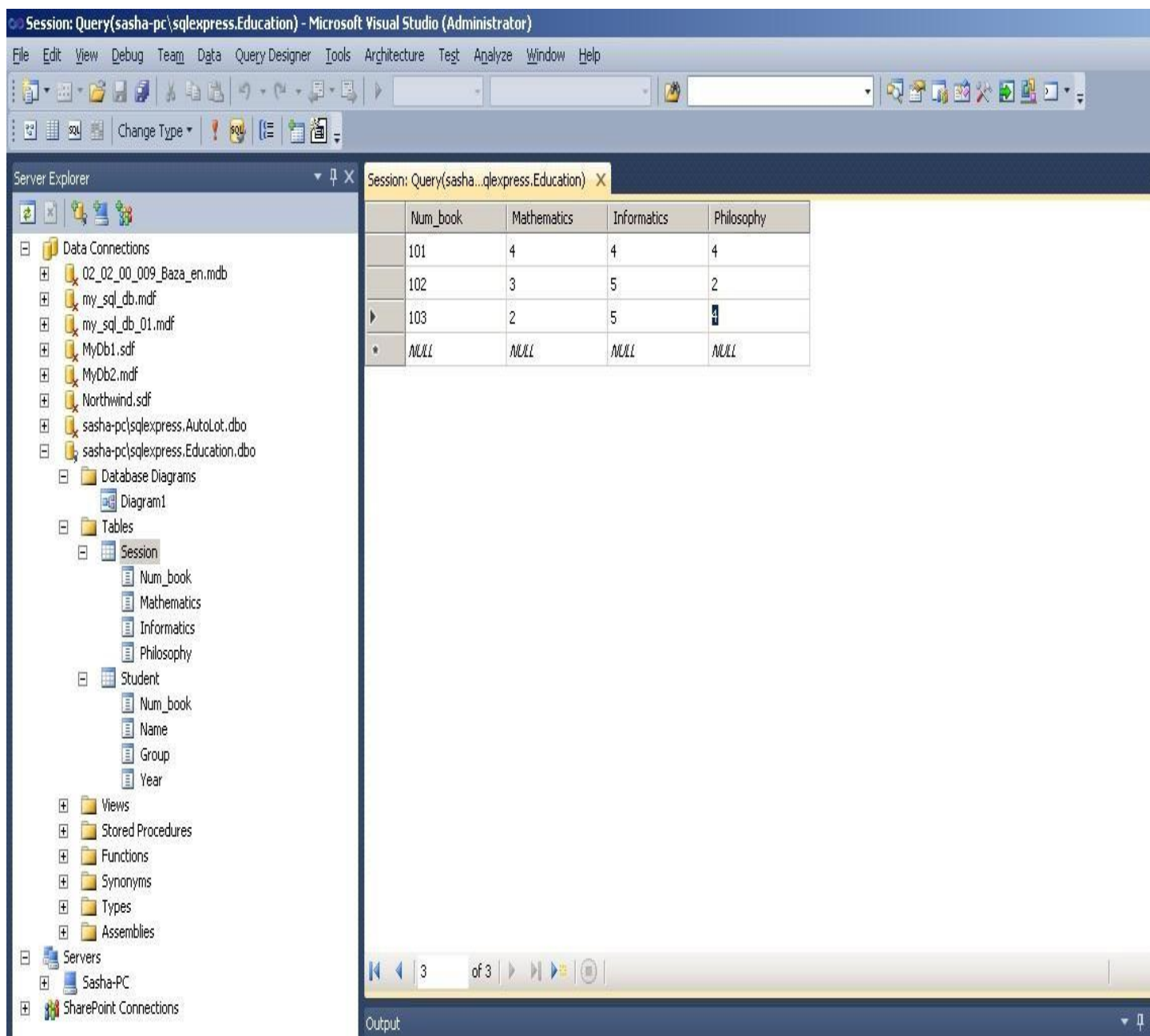


Рис. 2.26. Таблица Session с введенными данными

Итак, база данных создана. Ввод и обработку данных в таблицах можно реализовать программным путем.

3. Перенос файла БД Microsoft SQL на другой компьютер (Обращаю Ваше внимание-это общая технология, в конкретной версии Visual Studio могут быть вариации)

В большинстве случаев необходимо разрабатывать приложения, использующие в качестве базы данных Microsoft SQL Server. Наиболее рациональным решением является разработка базы данных в формате Microsoft SQL на рабочем компьютере с установленной локальной версией Microsoft SQL Server.

При сдаче проекта заказчику возникает необходимость переноса базы данных с локального компьютера. Для переноса на другой компьютер нам потребуется скопировать два файла – саму базу данных имя вашей базы.mdf и файл отчетов о транзакциях имя вашей базы .ldf. Однако непосредственное копирование данных файлов невозможно, так как данные файлы используются сервером баз данных. Для того чтобы сделать файлы доступными для копирования, базу данных необходимо отсоединить от сервера через диалоговое окно «Отсоединение базы данных». Подтверждаем отсоединение, нажимая кнопку «ОК», – и база отсоединена. Теперь нужные файлы доступны для копирования.

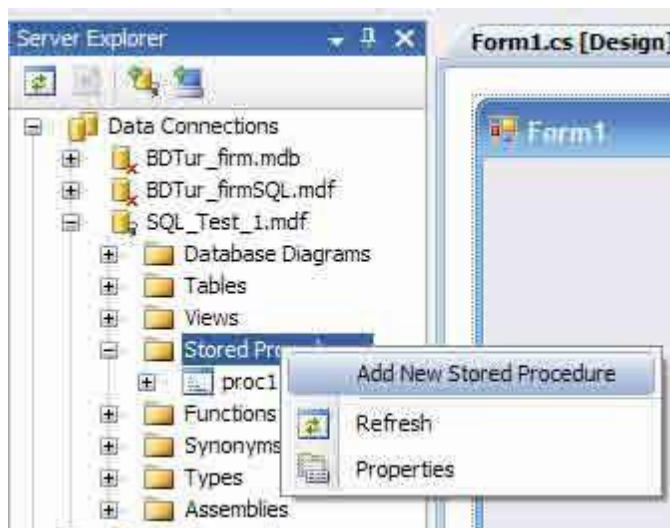
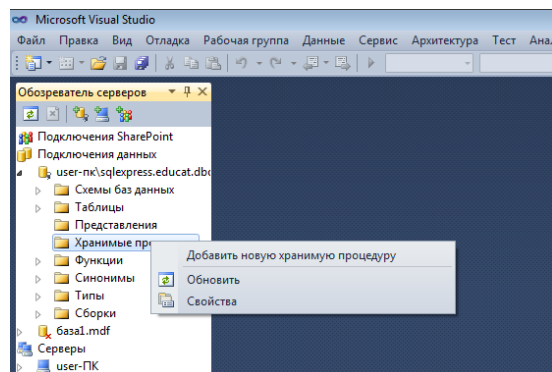
Для присоединения базы данных на другом компьютере запускаем Visual Studio, выделяем ветку «Базы данных» и в контекстном меню выбираем «Присоединить».

В появившемся окне указываем расположение файла базы данных Имя вашей базы.mdf – файл отчетов присоединится автоматически – и нажимаем «ОК». Присоединившаяся база данных немедленно отображается в папке «Базы данных». Следует отметить, что после присоединения БД может потребоваться настройка пользователей БД и прав доступа.

Создание хранимых процедур в базе данных при наличии подключения к ней

Среда Visual Studio предоставляет интерфейс для создания хранимых процедур в базе данных при наличии подключения к ней.

Запустим Visual Studio (даже нет необходимости создавать какой-либо проект), перейдем на вкладку «Обозреватель баз данных» («Server Explorer»), раскрываем подключение к базе данных, затем на узле «Хранимые процедуры» («Stored



Procedures») щелкаем правой кнопкой и выбираем пункт «Добавить новую хранимую процедуру» («New Stored Procedure»)

Рис. 3.1. Создание новой процедуры в окне «Server Explorer» Появляется шаблон структуры, сгенерированный мастером:

```
CREATE PROCEDURE dbo.StoredProcedure1
```

```
/*(
```

```
@parameter1 int = 5, @parameter2
```

```
datatype OUTPUT
```

```
)
```

```
*/ AS
```

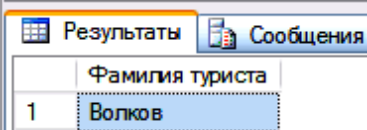
```
/* SET NOCOUNT ON */RETURN
```

Для того чтобы приступить к редактированию, достаточно убрать знаки комментариев «/*», «*/».

Команда NOCOUNT со значением ON отключает выдачу сообщений о количестве строк таблицы, получающейся в качестве запроса. Необходимость данного оператора заключается в том, что при использовании более чем одного оператора (SELECT, INSERT, UPDATE или DELETE) в начале запроса надо поставить команду SET NOCOUNT ON, а перед последним оператором SELECT – команду SET NOCOUNT OFF.

Например, хранимую процедуру proc_po1 (см. таблицу 3.1)

Таблица 3.1

Описание	
Извлечение фамилии туриста по заданному коду	
SQL-конструкция для создания	Команда для извлечения
create proc proc_po1 @TouristID int, @LastName nvarchar(60) output as select @LastName = Фамилия from Туристы where [Код туриста] = @TouristID	declare @LastName nvarchar(60) exec proc_po1 '4', @LastName output select @LastName as 'Фамилия туриста'
Результат запуска	
	

можно переписать следующим образом:

```
CREATE PROCEDURE dbo.proc_vs1(  
@TouristID int,  
@LastName nvarchar(60) OUTPUT  
) AS  
SET NOCOUNT ON
```

RETURN

«Разработать блок SQL» («Design SQL Block»), можно перейти к построителю запросов (Query Builder) (рис. 3.2).

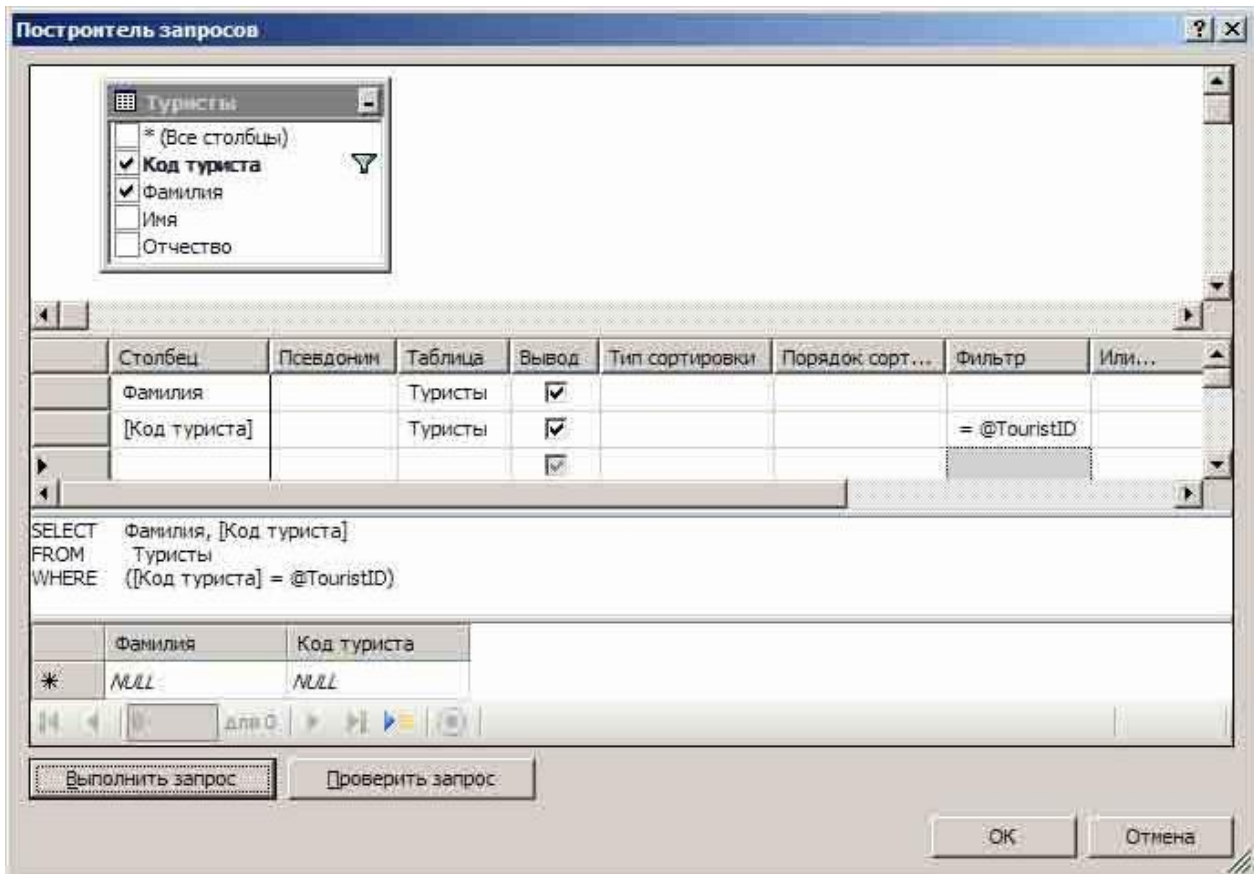


Рис. 3.2. Построитель запросов (Query Builder)

Для выполнения процедуры необходимо в выпадающем меню процедуры, представленном на рисунке 3.3, выбрать пункт Execute.

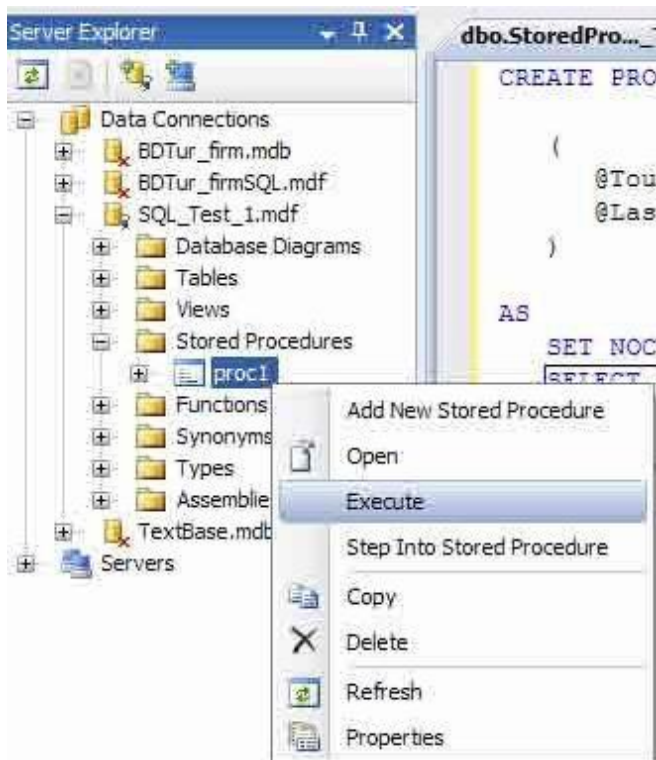


Рис. 3.3. Выполнение хранимой процедуры

При попытке выполнить хранимую процедуру в окно сообщений будет выведена информация о том, что данная процедура требует значение параметра. Для записи процедуры в базу данных достаточно ее сохранить. Происходит синхронизация с базой данных, и процедура "proc_vs1" появляется в списке. Двойной щелчок открывает ее для редактирования, причем заголовок имеет следующий вид:

`ALTER PROCEDURE dbo.proc_vs1`

Оператор ALTER позволяет производить действия (редактирование) с уже имеющимся объектом базы данных.

4. Работа с базами данных. технология ADO.NET.

4.1. Создание приложений баз данных

Реализация клиентского приложения с помощью проекта на языке C#

В среде Microsoft Visual Studio работа по созданию клиентского приложения начинается с построения проекта для выбранного языка программирования, который выбирается в окне, открывающемся после исполнения команды среды File/New/Project и создания нового проекта Windows Application

4.2. Пример простейшего приложения баз данных

Создадим простое приложение баз данных, которое выводит на экранную форму информацию из таблицы «Туристы» и связанную с текущей записью таблицы «Туристы» запись таблицы «Информация о туристах» из базы данных

Для этого создадим пустое Windows-приложение1. Внешний вид среды разработки приведен на рисунке 4.1.

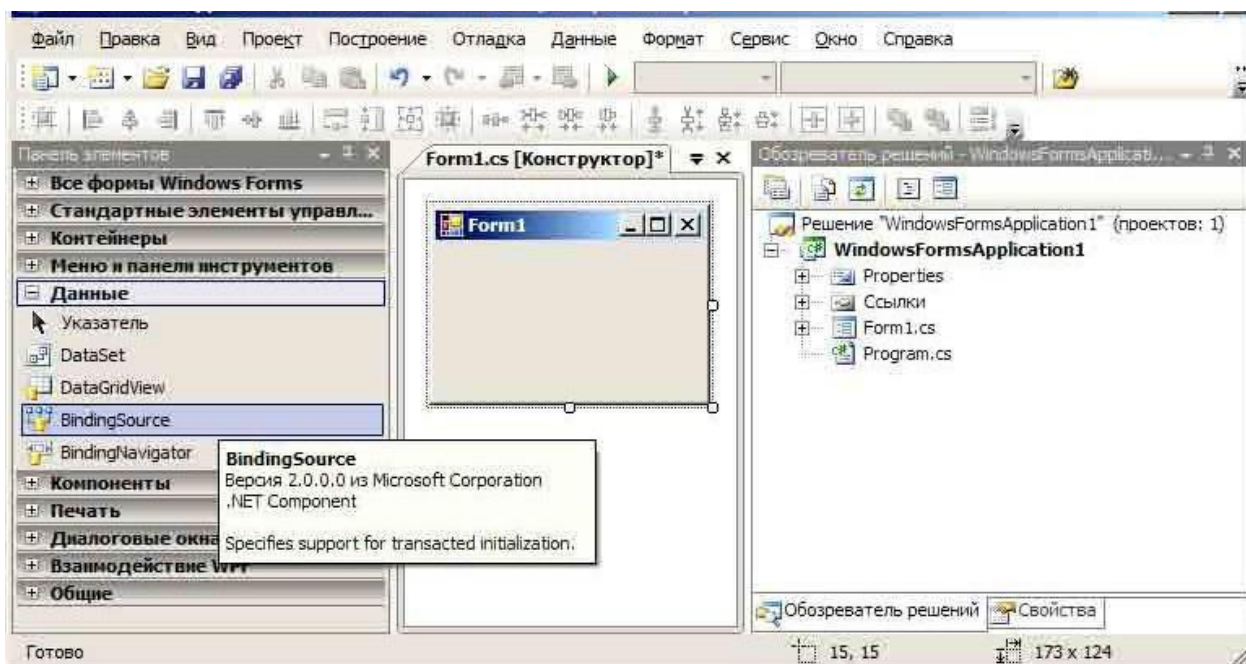


Рис. 4.1. Пустое приложение

На рисунке 4.1 выделена группа компонентов «Данные» («Data»), которая содержит компоненты для доступа к данным и манипулирования ими.

Привязку данных БД к форме осуществляет компонент «Binding Source».

Перенесем его на форму. После размещения его на форме среда разработки принимает следующий вид (рис. 4.2).

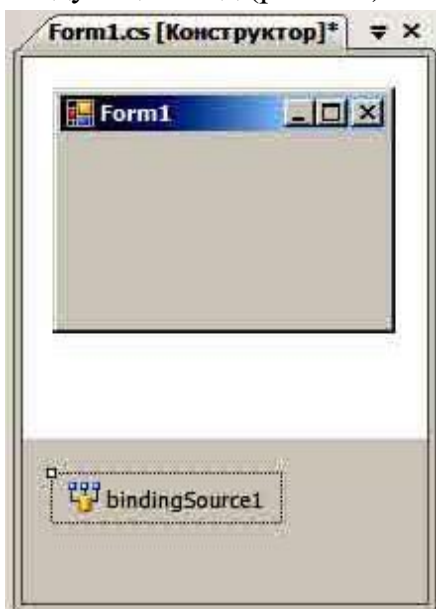


Рис. 4.2. Компонент Binding Source на форме

Компонент является не визуальным, поэтому он отображается на дополнительной панели. Основным свойством компонента является свойство DataSource, указывающее на источник данных. По умолчанию свойство является пустым, поэтому необходимо сформировать его значение. При выборе данного свойства в окне свойств появляется

следующее окно (рис. 4.3).

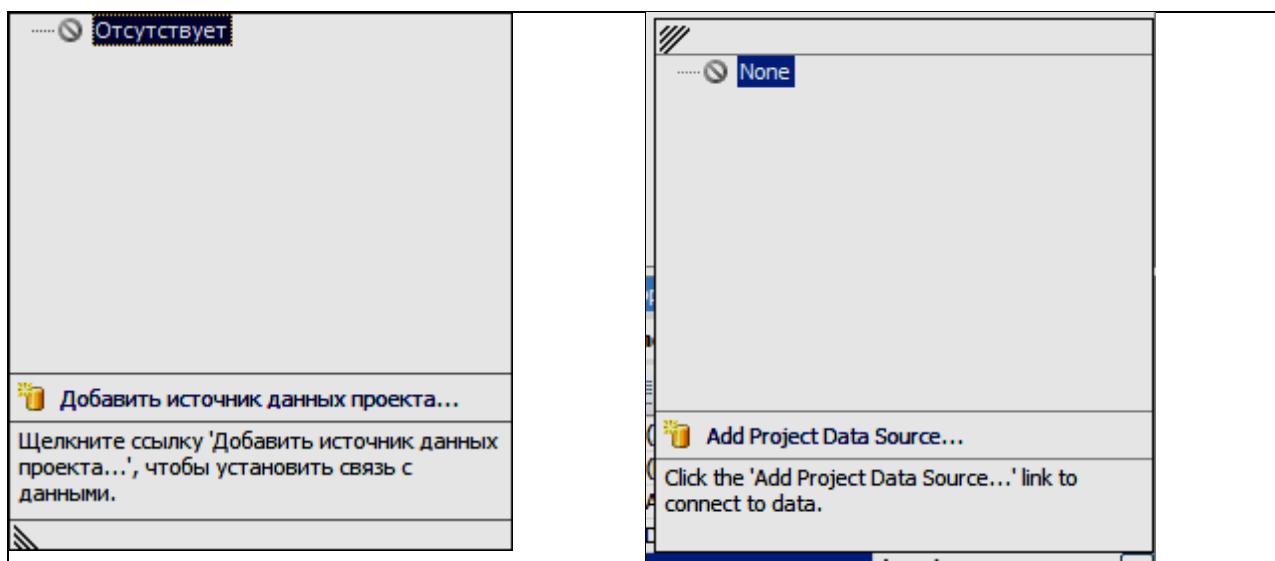


Рис. 4.3. Список источников данных. Компонент «Binding Source»

В настоящий момент список пуст, поэтому необходимо создать новый источник данных, выбрав команду «Add Project Data Source» для создания нового источника данных и соединения с ним. Появляется следующее окно диалога (рис. 4.4).

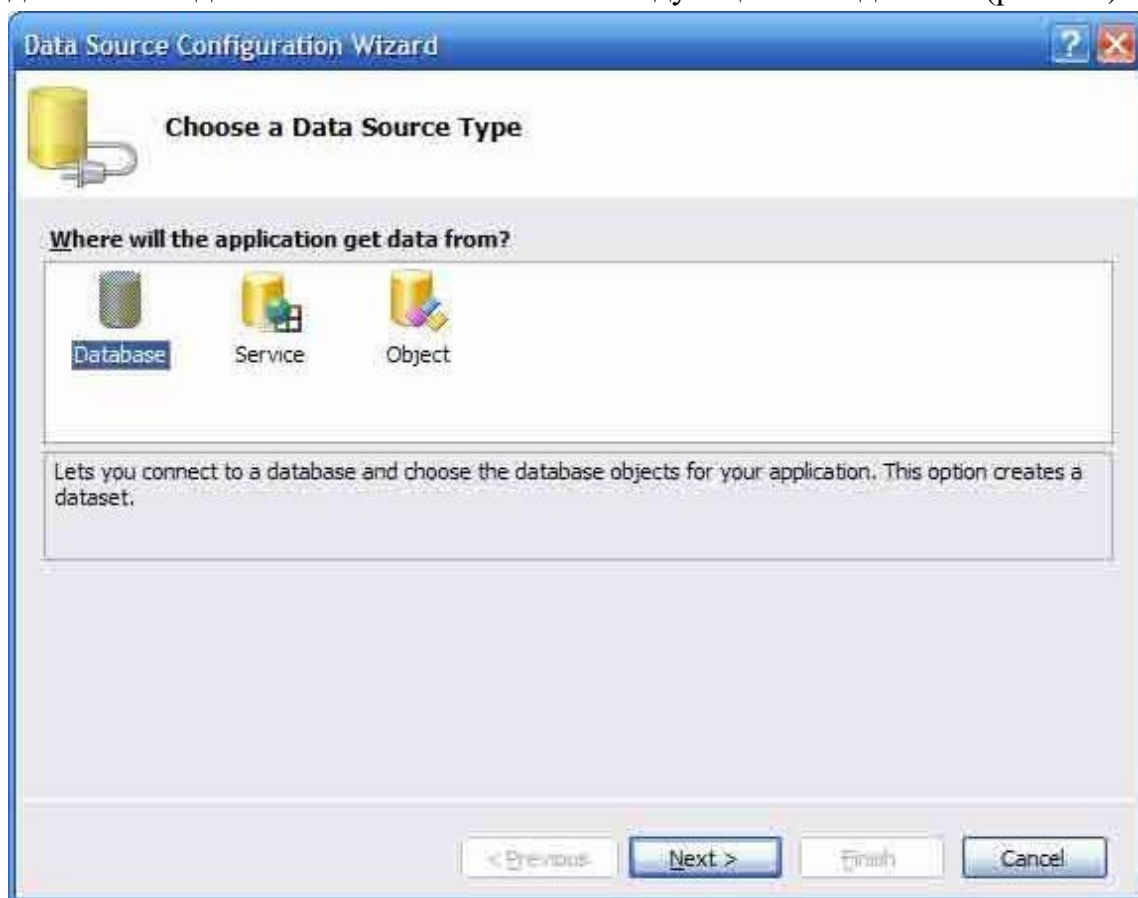


Рис. 4.4. Список источников данных

Данный диалог предоставляет следующий выбор источников данных:

- Database – База данных;
- Service – Служба, это некоторый сервис, предоставляющий данные. Чаще всего это Web-сервис;
- Object – Объект для выбора объекта, который будет генерировать данные и объекты для работы с ними.

В нашем случае необходимо выбрать пункт «База данных» («Database»). Появляется окно выбора соединения с данными (рис. 4.5).

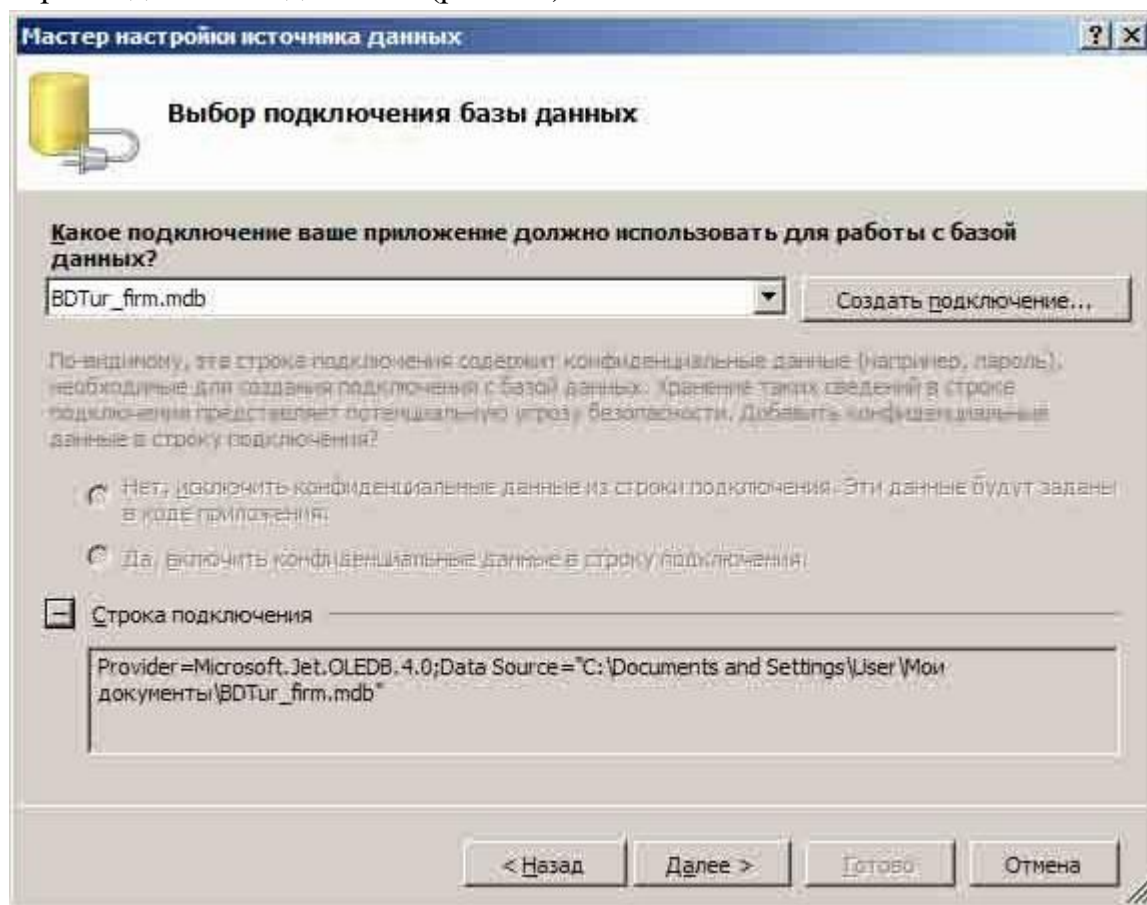


Рис. 4.5. Выбор соединения с данными

Целью данного диалога является создание строки соединения, в которой будут описаны параметры соединения для механизма ADO, такие как тип базы данных, ее местонахождение, имена пользователей, средства безопасности и пр.

В выпадающем списке диалога находятся все создаваемые ранее соединения. Если необходимого соединения в списке нет, то следует использовать кнопку «Создать подключение» («New connection»). Нажатие кнопки приводит к появлению следующего диалога (рис. 4.6).

В данном диалоге выбирается тип источника данных (в данном случае Microsoft Access), имя базы данных (в данном случае имя и местоположение файла базы данных), имя пользователя и пароль, используемые для подключения к базе данных. Кнопка «Дополнительно» («Advanced») позволяет задать большое количество параметров, относящихся к различным деталям механизма ADO. Использование кнопки «Проверить

подключение» («Test Connection») позволит убедиться в правильности введенных параметров и работоспособности соединения.

Следующий шаг диалога предлагает сохранить полученную строку соединения в файле настроек приложения. Рекомендуется принять данный выбор для упрощения последующего размещения и поддержки программного продукта.

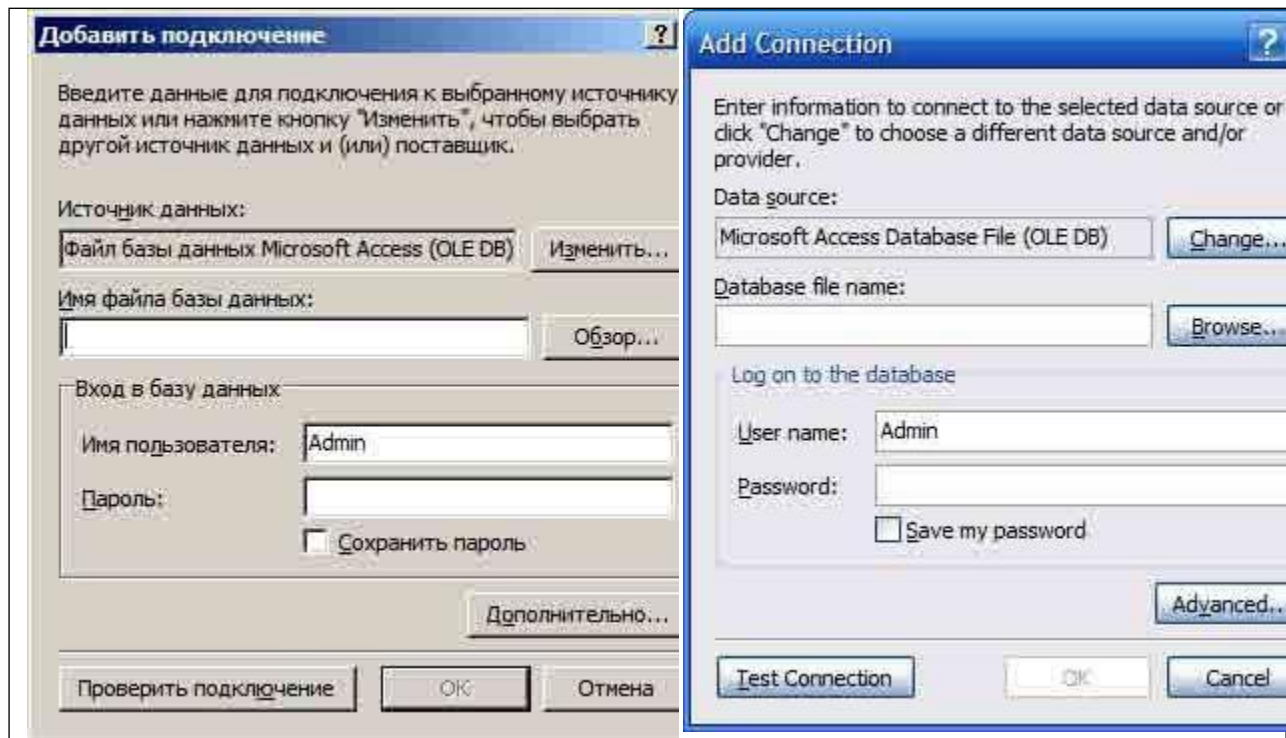


Рис. 4.6. Создание нового соединения

Последний шаг диалога – выбор тех таблиц или иных объектов базы данных, которые необходимы в данном источнике данных. Окно выбора представлено на рисунке 4.7.

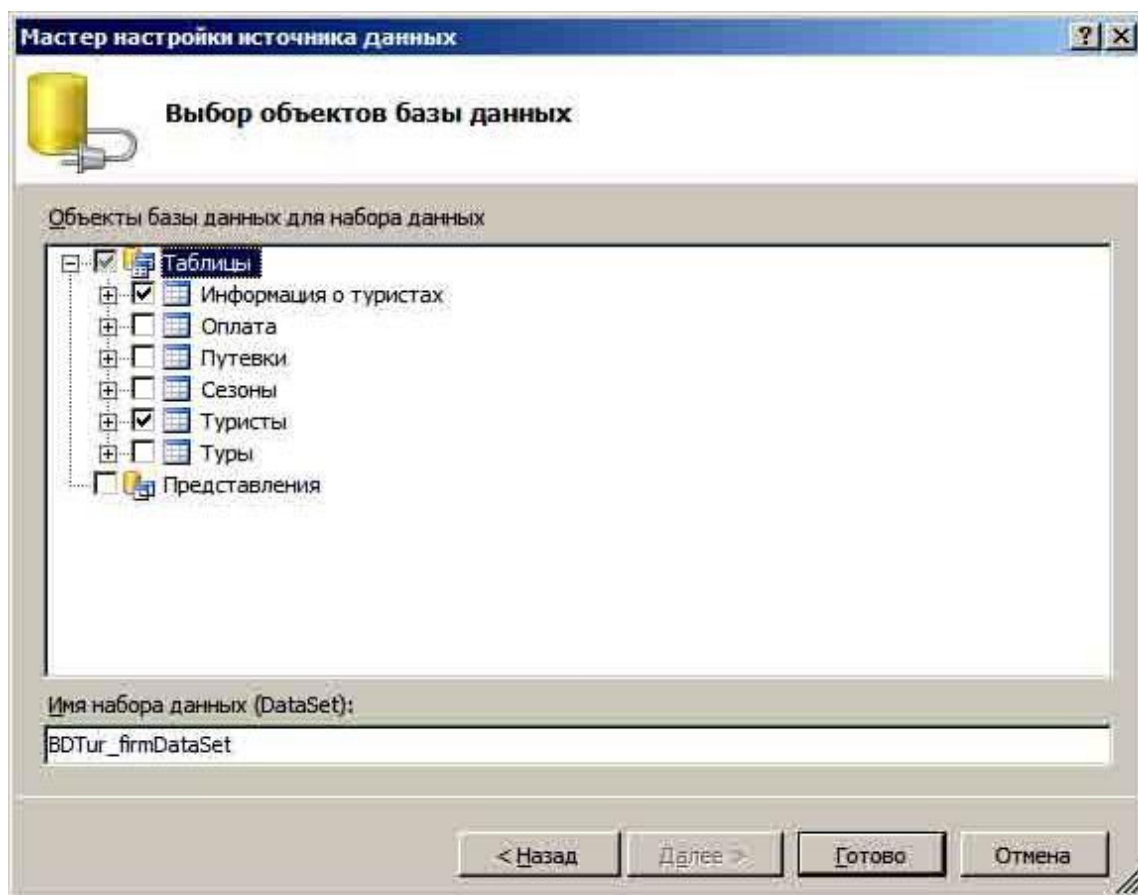


Рис. 4.7. Выбор необходимых таблиц

В данном окне выбраны таблицы «Туристы» и «Информация о туристах».

Поскольку иных объектов, кроме таблиц, в базе данных не было создано, на рисунке 4.7 отображаются только таблицы. На этом создание источника данных завершено. После нажатия кнопки «Готово» («Finish») рядом с компонентом BindingSource на форме появляется компонент DataSet.

Теперь данные, подключенные выше, необходимо отобразить на форме. Простейшим способом отображения данных является использование компонента DataGridView из группы компонентов Data. Компонент является визуальным и на форме выглядит следующим образом (рис. 4.8).

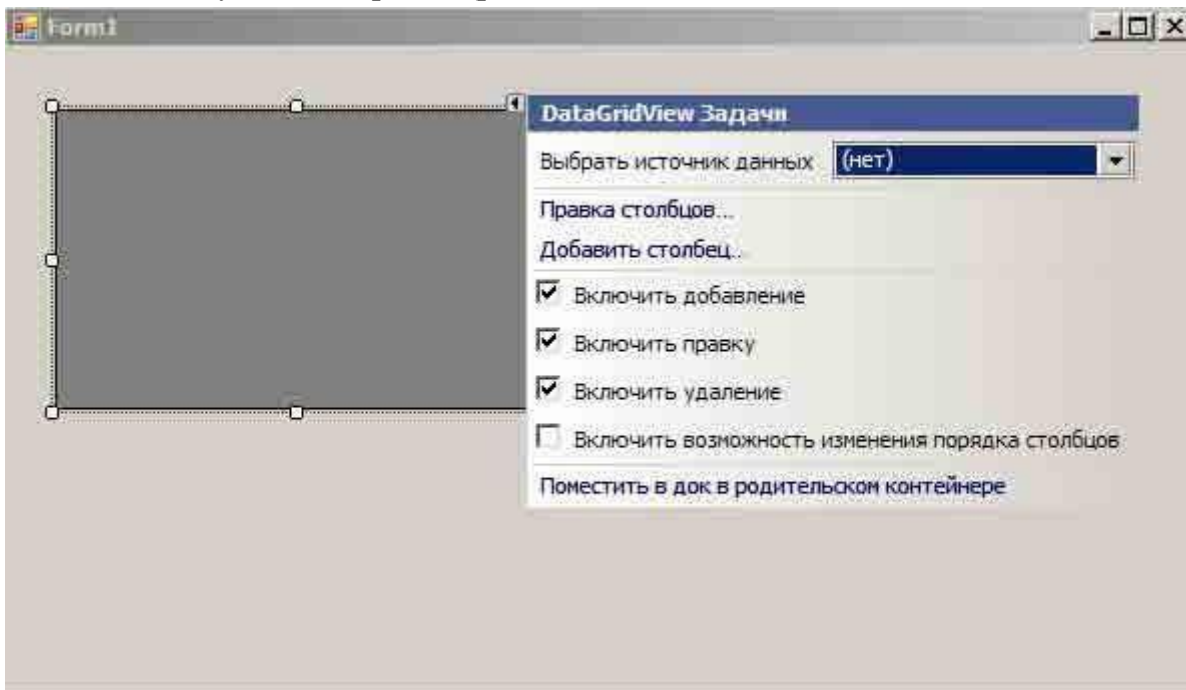


Рис. 4.8. Компонент DataGridView

Сразу же возникает окно настройки компонента, которое определяет его возможности по редактированию данных: «Включить редактирование» («Enable Adding»), «Включить правку» («Enable Editing»), «Включить удаление» («Enable Deleting»); возможность изменения последовательности столбцов:

«Включить возможность изменения порядка столбцов» («Enable Column Reordering»); а также возможность закрепления в контейнере-родителе.

Для того чтобы компонент мог отображать данные, необходимо выбрать источник данных в выпадающем списке. Выбор выпадающего списка приводит к появлению следующего диалога (рис. 4.9).

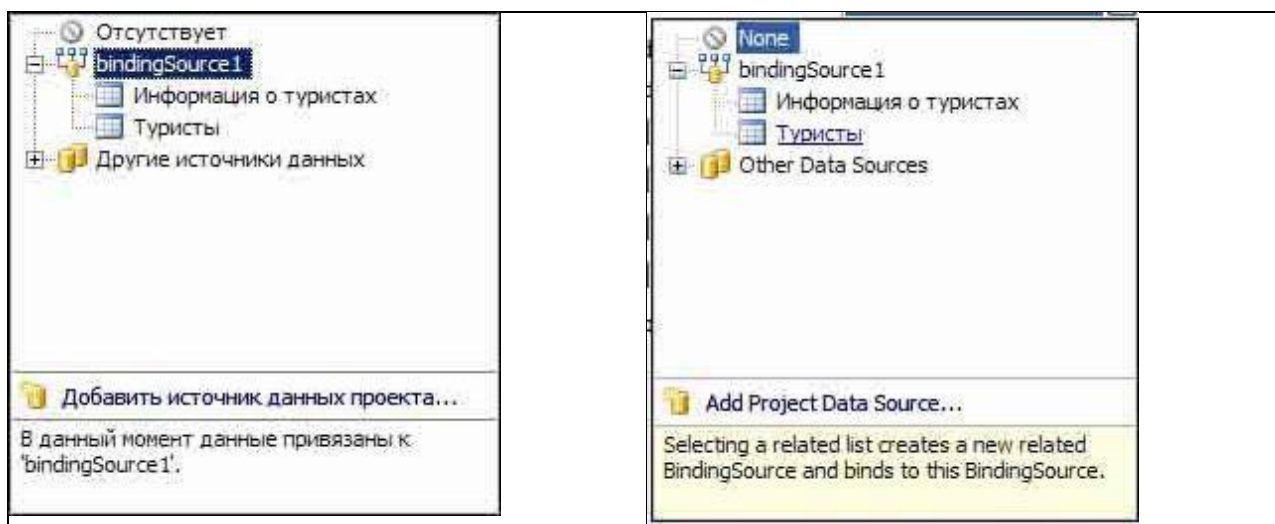


Рис. 4.9. Выбор источника данных для DataGridView

В данном случае мы выбрали в качестве источника данных таблицу «Туристы». Данный выбор изменяет экранную форму следующим образом (рис. 4.10).

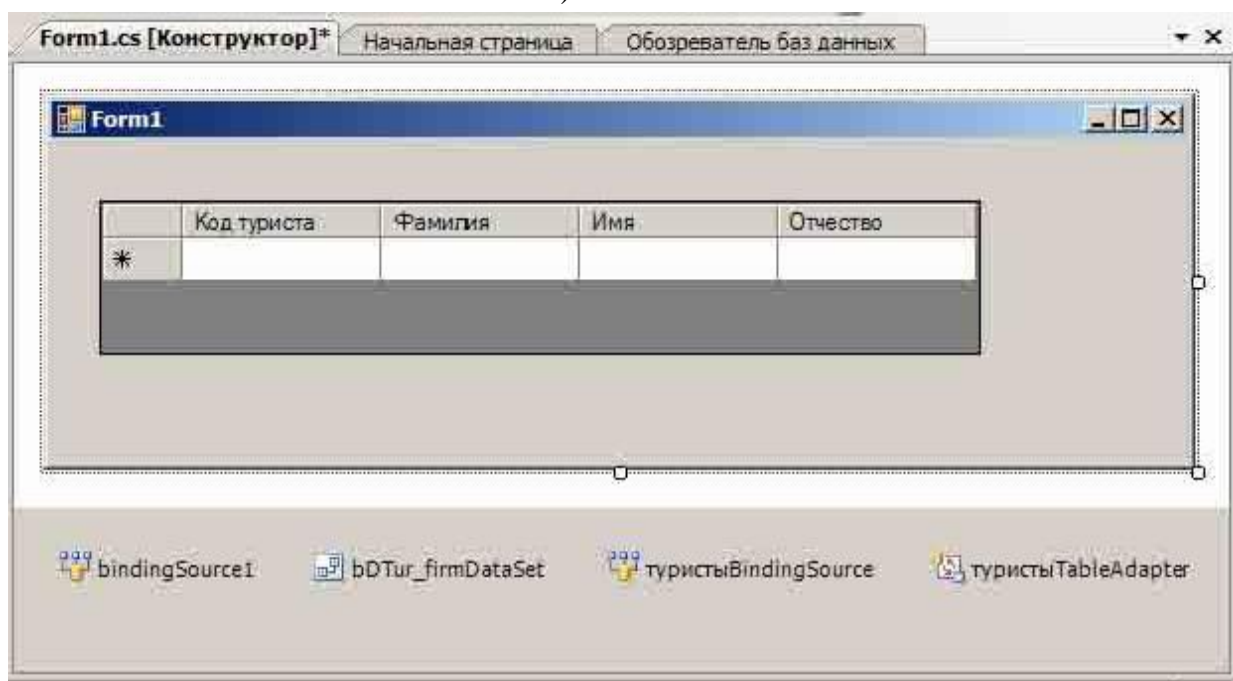


Рис. 4.10. Компонент DataGridView отображает структуру таблицы

На рисунке видно, что появился еще один компонент BindingSource и компонент TableAdapter, работающий с таблицей «Туристы». Обратите внимание, что в design-time или в процессе разработки данные из таблицы не отображаются.

Теперь необходимо отобразить данные из связанной таблицы «Информация о туристах». Для этого разместим на форме еще один компонент DataGridView и в качестве источника данных выберем следующее (рис. 4.11).

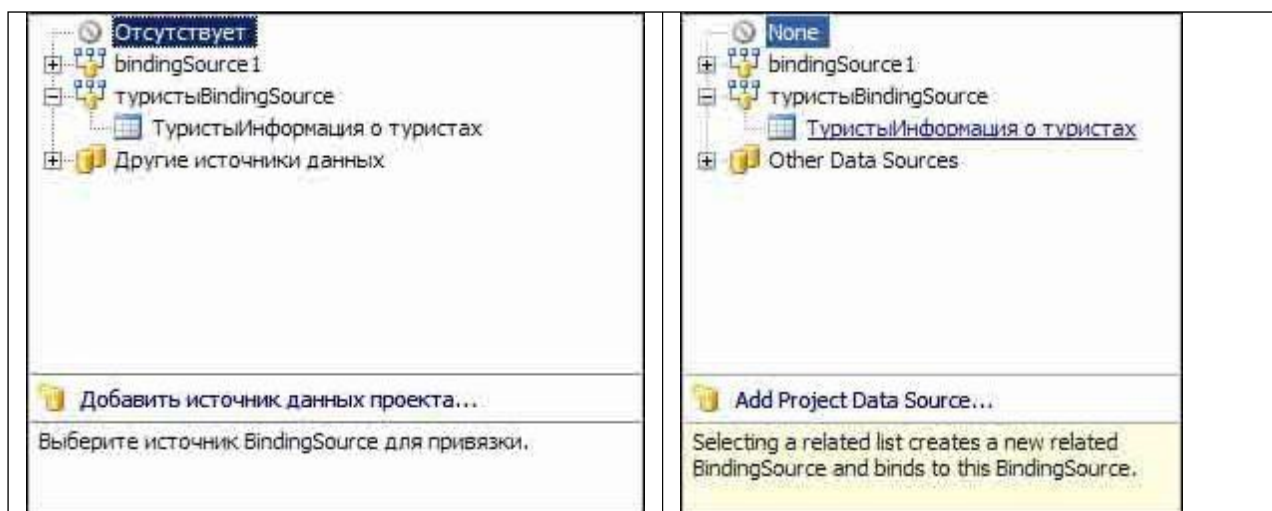


Рис. 4.11. Выбор источника данных для второго DataGridView

Здесь в качестве источника данных выступает не сама таблица «Информация о туристах», а связь (Binding Source) между таблицами «Туристы» и «Информация о туристах». Такой выбор гарантирует выбор из таблицы «Информация о туристах» только тех строк, которые связаны с текущей строкой в таблице «Туристы». Также такой выбор гарантирует правильность обновления и удаления связанных данных. Работа полученного приложения показана на рисунке 4.12.

Код туриста	Фамилия	Имя	Отчество
1	Иванов	Василий	Степанович
2	Николаев	Олег	Валентинович
3	Андреева	Инна	Вячеславовна
4	Волков	Антон	Павлович
5	Кириллова	Ольга	Михайловна
*			

Код туриста	Серия паспорта	Город	Страна	Телефон	Индекс
2	TE 1562487	Ростов	Россия	3216547	120035
*					

Рис. 4.12. Приложение базы данных в работе

Перемещение по данным при помощи стрелочных клавиш является неудобным. Для упрощения навигации по данным существует компонент BindingNavigator. Поместим его на форме (рис. 4.13).

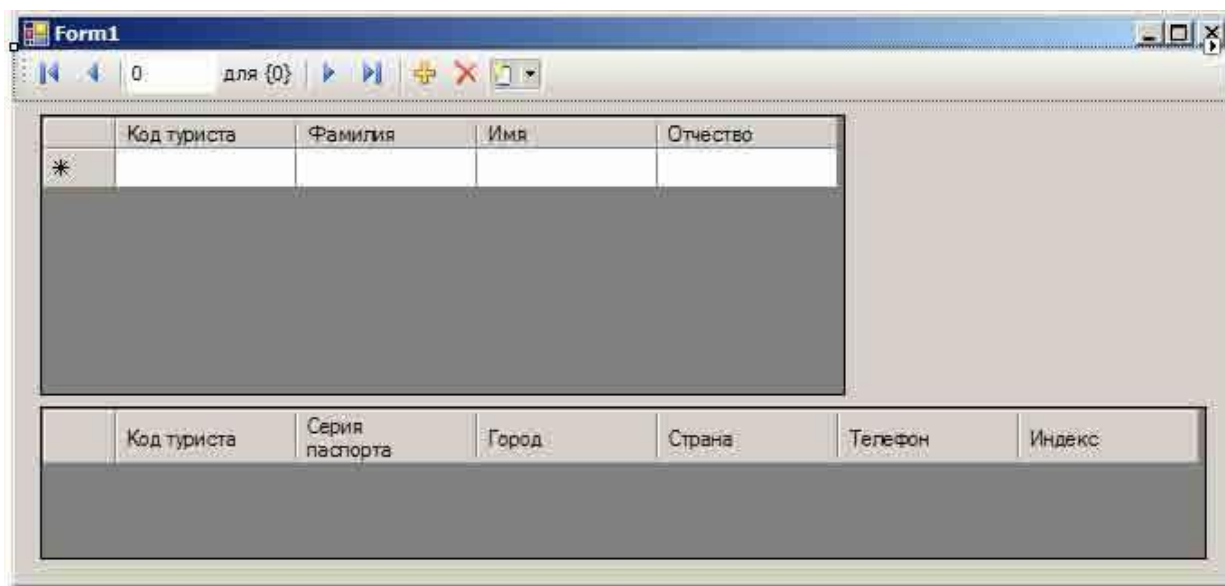


Рис. 4.13. Компонент BindingNavigator на форме

Данный компонент позволяет осуществлять навигацию между записями таблицы, добавлять и удалять строки таблицы. Возможности и внешний вид компонента можно настраивать, так как он представляет собой полосу меню ToolStripContainer. Свойством, определяющим таблицу, по которой производится навигация, является свойство DataSource. Установим значение этого свойства равным «туристыBindingSource». В работе компонент выглядит следующим образом (рис. 4.14).

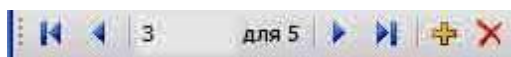


Рис. 4.14. Компонент BindingNavigator в работе

Редактирование данных в ячейках компонента DataGridView при соответствующих настройках возможно, но неудобно и не рационально. В частности, трудно проверять введенные значения на ошибки. Поэтому для таблицы «Туристы» сделаем экранную форму, позволяющую отображать данные в компонентах TextBox и редактировать их. Для этого разместим на форме контейнер типа Panel, а на нем три компонента TextBox следующим образом (рис. 4.15).

Рис. 4.15. Экранная панель для редактирования записей таблицы «Туристы»

Теперь необходимо осуществить привязку компонентов TextBox к соответствующим полям таблицы «Туристы». Для этого используем свойство из группы DataBindings – Advanced, показанное на рисунке 4.16.

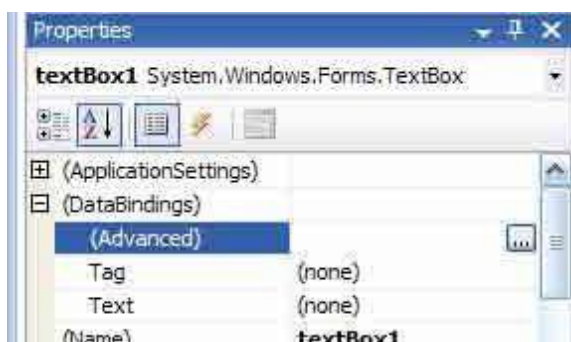


Рис. 4.16. Свойство «DataBindings – Advanced»

Выбор данного свойства приводит к появлению диалога, показанного на рисунке 4.17. Данный диалог позволяет осуществить не только привязку данных, но также задать событие, в рамках которого будет проводиться обновление данных, а также форматирование данных при их выводе.

Для верхнего компонента TextBox в выпадающем списке Binding выберем источником данных «туристыBindingSource» и поле источника – «Фамилия».

Для среднего и нижнего компонентов TextBox выберем тот же источник данных и поля «Имя» и «Отчество» соответственно.

Разработанное приложение в работе выглядит следующим образом (рис. 4.18).

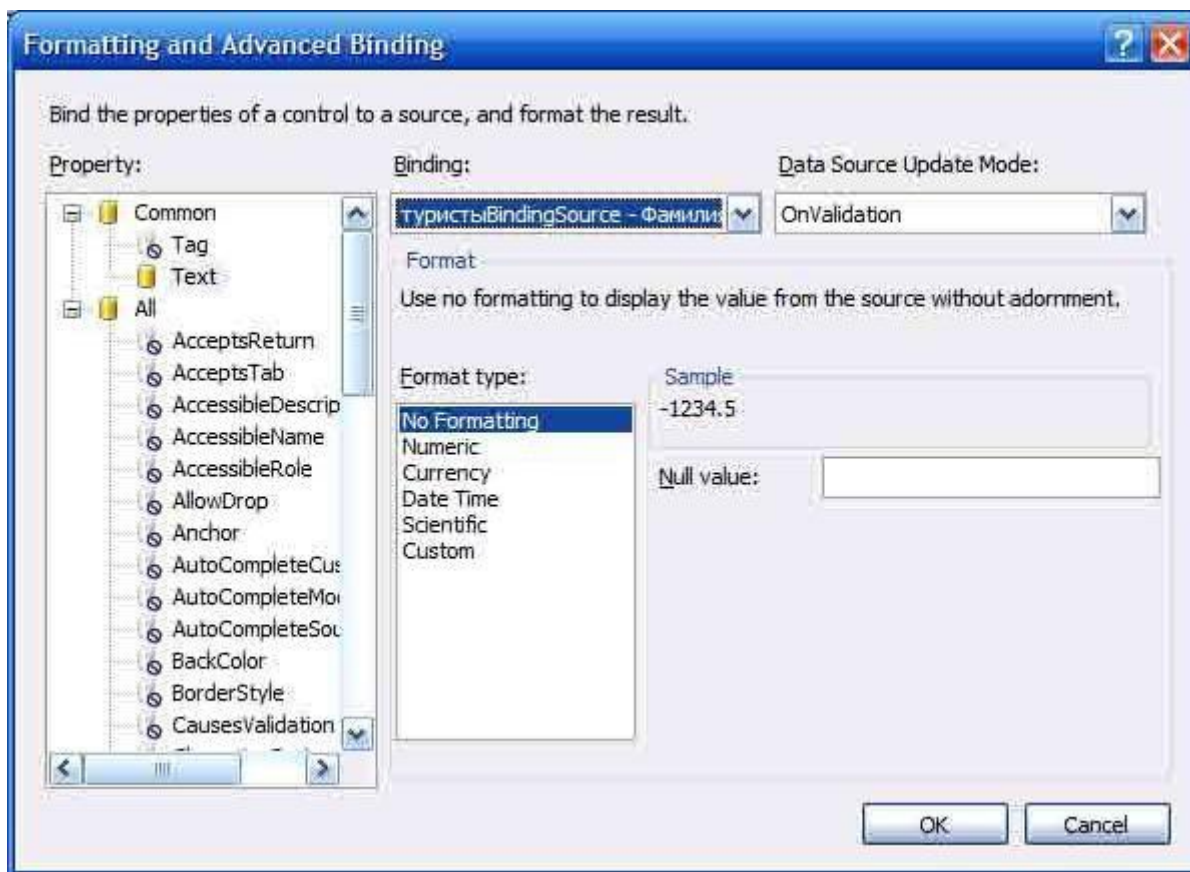


Рис. 4.17. Окно диалога для свойства «DataBindings – Advanced»

Код туриста	Фамилия	Имя	Отчество
1	Иванов	Василий	Степанович
2	Николаев	Олег	Валентинович
3	Андреева	Инна	Вячеславовна
4	Волков	Антон	Павлович
5	Кириллова	Ольга	Михайловна
*			

Код туриста	Серия паспорта	Город	Страна	Телефон	Индекс
4	CA 1869742	Москва	Россия	1234500	650378
*					

Волков
Антон
Павлович

Рис. 4.18. Привязка данных к визуальным компонентам

Однако при внесении изменений все новые данные остаются только на форме. В базе данных они не сохраняются, и при повторном вызове приложения, конечно же, будут отсутствовать. Это происходит потому, что данные были загружены в объект DataSet, который представляет собой копию таблицы в памяти. Все

действия выполняются с этой копией. Для того чтобы изменения отображались в базе данных, необходимо выполнить метод Update класса TableAdapter. Таким образом, в разрабатываемом приложении необходимо разместить кнопку «Обновить» и записать в обработчик события Click следующий программный код:

```
туристыTableAdapter.Update(bDTur_firmDataSet);
информация_o_туристахTableAdapter.Update(bDTur_firmDataSet);
```

Данный код обновляет информацию в таблицах «Туристы» и «Информация о туристах», предоставляемых источником данных. Отметим, что данный метод является перегруженным, и его варианты позволяют обновлять как отдельную строку таблицы, так и группу строк.

4.3. Создание приложения БД «вручную»

Выше было рассмотрено создание приложений для работы с базами данных с использованием различных компонентов визуальной среды. Однако разрабатывать такие приложения можно также без использования визуальной среды.

Пример создания приложения БД «вручную»

Создадим в параллельно два приложения для работы с созданными ранее базами данных, аналогичные приложению, рассмотренному выше (см. п. 4.1).

Работа будет проводиться, соответственно, со следующими таблицами:

- Microsoft Access – BDTur_firm.mdb ;
- Microsoft SQL – BDTur_firmSQL.mdf .

Начнем с запуска Visual Studio и создания нового проекта Windows Application. Размещаем на создавшейся форме элемент управления DataGridView, свойству Dock которого устанавливаем значение Fill. Переходим в код формы и подключаем соответствующие пространства имен:

- для MS Access – using System.Data.OleDb;
- для MS SQL – using System.Data.SqlClient;

В любом случае необходимо подключить пространство имен System.Data.

В конструкторе формы после InitializeComponent создаем объект DataAdapter.

В приложении, работающем с MS Access, соответствующий код будет выглядеть следующим образом:

```
public Form1()
{
    //
    // Required for Windows Form Designer support
    //
    InitializeComponent
```

```

nt();
OleDbDataAdapter dataAdapter =
new OleDbDataAdapter(CommandText, ConnectionString);
}

```

А в приложении, работающем с MS SQL:

```

public Form1()
{
//
// Required for Windows Form Designer support
//
InitializeComponent();
SqlDataAdapter dataAdapter =
new SqlDataAdapter(CommandText, ConnectionString);
}

```

Как видно из приведенного кода, фрагменты отличаются только названиями объектов. В качестве параметров DataAdapter передаются CommandText и ConnectionString. Переменная типа string CommandText представляет собой обычный SQL-запрос на выборку из таблицы «Туристы», а переменная типа ConnectionString – это так называемая строка подключения, в которой указываются расположение базы данных, ее название, параметры авторизации и пр. Воспользуемся следующими строками подключения и командами:

```

// MS Access
CommandText: "SELECT [Код туриста], Фамилия, Имя, Отчество FROM
Туристы"; ConnectionString: "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source="D:\БМИ\For ADO\BDTur_firm.mdb"

```

```

// MS SQL:
CommandText: "SELECT CustomerID, CompanyName, ContactName,
ContactTitle, Address, City, Region, PostalCode, Country, Phone, Fax FROM
Customers"
ConnectionString: "Data
Source=.\\SQLEXPRESS;AttachDbFilename="D:\БМИ\For ADO\
BDTur_firmSQL.mdf";Integrated Security=True;Connect Timeout=30;User
Instance=True"

```

Обратите внимание на названия переменных CommandText и ConnectionString. Когда создается объект DataAdapter, в качестве параметров можно передать

названия строк, таких как `cmdText` и `conString`, или даже `cmt` и `cns` – совершенно равноправно, не забыв, конечно же, назвать также эти переменные в классе `Form1`. Но сама среда Visual Studio генерирует эти строки именно с такими названиями – `CommandText` и `ConnectionString`, поэтому такое название переменных облегчает поддержку и сопровождение разработанного программного продукта.

Продолжим создание программы. Дальнейший код одинаков для обоих вариантов. Создаем объект `DataSet`:

```
DataSet ds = new DataSet();
```

Заполняем таблицу «Туристы» объекта `ds` данными из базы:

```
dataAdapter.Fill(ds, "Туристы");
```

Связываем источник данных объекта `dataGridView1` с таблицей «Туристы» объекта `ds`:

```
dataGrid1.DataSource = ds.Tables["Туристы"].DefaultView;
```

Теперь запустим созданное приложение. Если все сделано правильно, то на экранной форме отобразится содержимое таблицы «Туристы».

Теперь изменим код следующим образом:

```
dataAdapter.Fill(ds, "Туристы2");
```

```
dataGridView1.DataSource = ds.Tables["Туристы2"].DefaultView;
```

Таблицы «Туристы2» в БД нет, однако код по-прежнему работает. Это связано с тем, что таблица, которую мы называем «Туристы», при вызове метода `Fill` объекта `dataAdapter` может быть названа как угодно – ее содержимое будет представлять собой извлекаемую таблицу из базы данных. При указании источника данных (`DataSource`) для объекта `dataGridView1` мы ссылаемся именно на таблицу «Туристы», которая была создана при вызове метода `Fill`. Таким образом, при заполнении таблиц их можно называть произвольным образом. Однако для реальных проектов рекомендуется использовать настоящие названия таблиц, чтобы избежать путаницы и трудностей в сопровождении.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основные источники:

1. Агальцов В.П. Базы данных. Распределенные и удаленные базы данных: учебник. – М.: ФОРУМ: ИНФРА-М, 2017. -271с.
2. Баженова И.Ю. Основы проектирования приложений баз данных / И.Ю. Баженова.
- 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016.
- 238 с.: ил. - Библиогр. в кн. - ISBN 5-94774-539-9; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428933>.

3. Базы данных в высокопроизводительных информационных системах: учебное пособие / авт.-сост. Е.И. Николаев; Министерство образования и науки РФ, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2016. - 163 с.: ил. - Библиогр.: с.161.; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=466799>.

4. Кузнецов С. Введение в реляционные базы данных / С. Кузнецов. - 2-е изд., исправ.

- Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 248 с.: ил. - (Основы информационных технологий). - Библиогр. в кн.; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429088>.

Дополнительные источники:

1. Баженова И.Ю. SQL и процедурно-ориентированные языки / И.Ю. Баженова. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 167 с.: ил. - Библиогр. в кн. - ISBN 5-94774-539-9; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428934>.

2. Гуров В.В. Архитектура и организация ЭВМ / В.В. Гуров, В.О. Чуканов. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 184 с.: ил., схем. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 5-9556-0040-X; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429021>.

3. Ковалев Д.В. Информационная безопасность : учебное пособие / Д.В. Ковалев, Е.А. Богданова; Министерство образования и науки РФ, Южный федеральный университет. - Ростов-на-Дону: Издательство Южного федерального университета, 2016. - 74 с.: схем., табл., ил. - Библиогр. в кн. - ISBN 978-5-9275-2364-1; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=493175>.

4. Кузнецов С. Введение в модель данных SQL: курс / С. Кузнецов. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 351 с.: ил. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 5-9556-00028-0; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429087>.

5. Кумскова И.А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

6. Мартишин С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2017. – 160 с.