**Создание дополнительных адаптеров**

При подключении базы к приложению MS VS, автоматически Мастером связи создается файл с расширением xsd, содержимым которого являются классы-адаптеры, связанные с таблицами подключенной базы данных. На рисунке 1 представлен пример содержимого файла xsd.

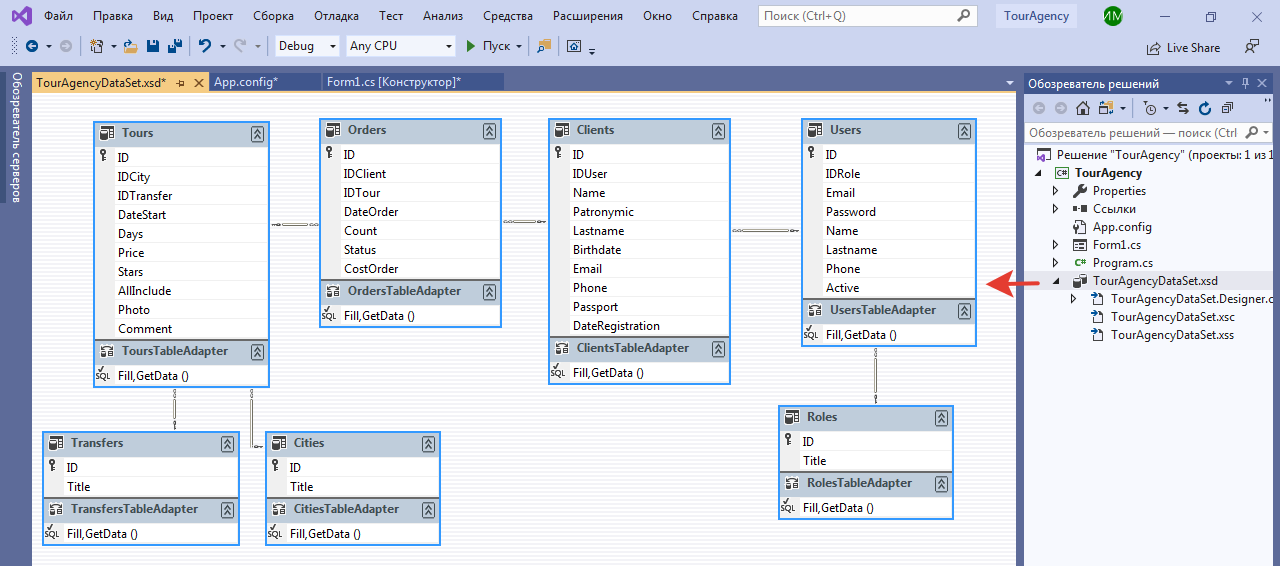


Рисунок 1 - Содержимого файла xsd

Полями этих классов являются поля таблицы БД. Для каждого класса-адаптера автоматически создаются два метода: Fill и GetData. В каждом из этих методов реализован запрос типа: SELECT \* FROM Таблица (получить все записи Таблиц) к той таблице, с которой связан класс-адаптер. Основная задача этих методов — перенос результата запроса из БД в оперативную память для дальнейшей обработки. В оперативной памяти создается объект типа DataSet. Отличие этих методов в результате и параметрах. В методе Fill объект типа DataSet находится в параметре метода и метод возвращает количество полученных записей из БД. В методе GetData объект типа DataSet возвращается как результат работы.

Часть возникает задача в создании дополнительных классов-адаптеров. Потребность в этом возникает, когда, например, строится запрос, в котором участвует сразу несколько таблиц подключенной БД. Рассмотрим алгоритм создания нового класса-адаптера и создадим для него метод, содержащий сложный запрос из нескольких таблиц.

**Создать новый класс-адаптер**

Открыть содержимое файла со схемой адаптеров (файл xds) двойным щелчком по нему в окне «Solution Explorer». Вызвать локальное меню окна схемы. Выбрать пункт локального меню «Добавить/TableAdapter…», как на рисунке 2.

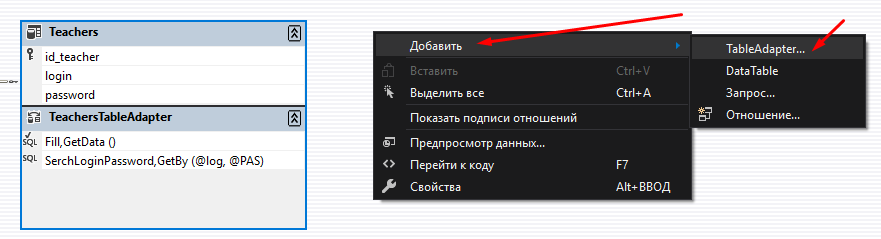


Рисунок 2 – Локальное меню окна схемы адаптеров

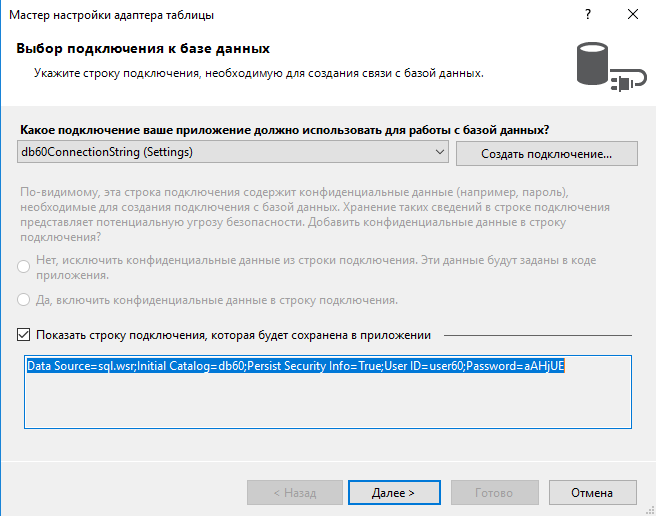
В открывшемся окне достаточно нажать кнопку «Далее», т.к. система уже настроила путь к БД.

Рисунок 3 – Выбор подключения

В следующем окне (см. рис. 4) предлагается выбрать тип создаваемой команды. Оставляем выбранной первую радиокнопку и выбираем кнопку «Далее».

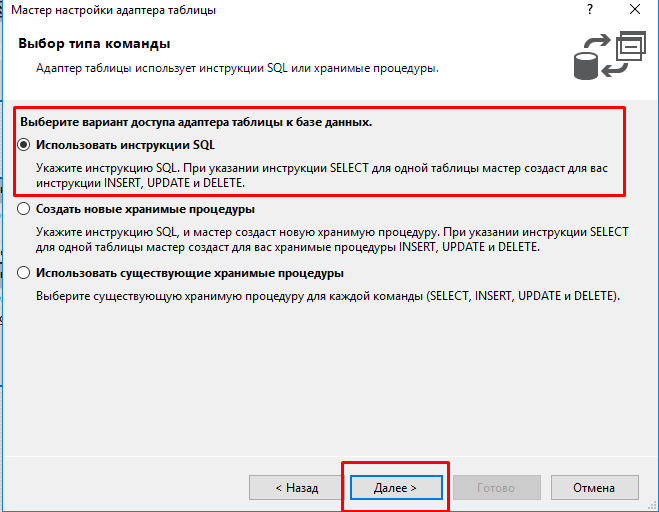


Рисунок 4 – Выбор типа создаваемой команды

В следующем окне надо построить SQL-запрос. Для этого вызываем Конструктор кнопкой «Конструктор запросов…» (см. рис. 5).

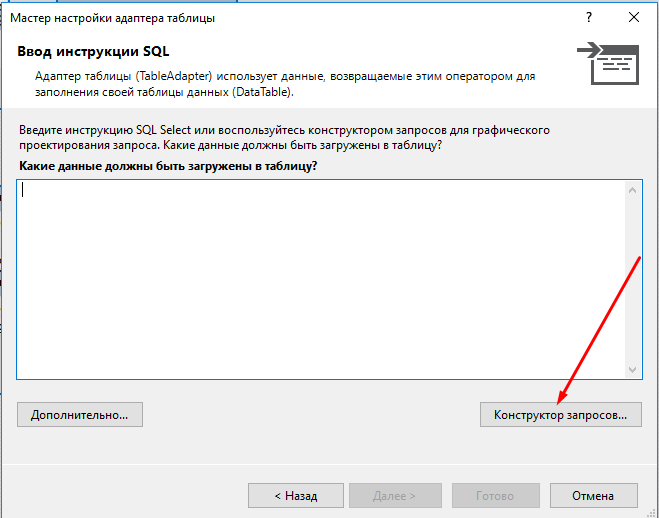


Рисунок 5 – Окно создания SQL-запроса

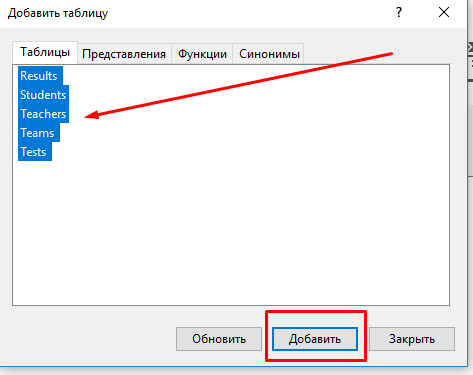
Появляется окно выбора таблиц из подключенной базы данных, которые будут участвовать в формируемом запросе. Для этого можно выбрать все таблицы, как показано на рисунке 6, чтобы использовать создаваемый класс-адаптер для создания нескольких методов с разными запросами. Можно выбрать часть таблиц. Выбрать кнопку «Добавить», чтобы все выбранные таблицы в окне 6 перенеслись в окно 7 – Конструктора запросов.

Рисунок 6 – Выбор таблиц для запроса

Сначала окно Конструктора запросов пустое. В верхней части находятся все те таблицы, которые были выбраны в предыдущем окне. Те поля тех таблиц, которые будут участвовать в запросе, должны быть отмечены устанавливаемыми флажками внутри каждой таблицы. Выбранные поля переносятся в среднюю часть окна, где на них можно наложить дополнительные фильтры, ограничения, вывод в запрос. В нижней части окна отображается строка формируемого запроса, учитывающая все связи между таблицами.

На рисунке 7 представлен запрос, созданный в Конструкторе запроса. Кнопкой «Выполнить запрос», можно проверить правильность работы созданного запроса. Результаты запроса отображаются в нижней части окна. Туда попадут те записи, которые подходят под условие созданного sql-запроса.

После проверки правильности работы запроса, выбираем кнопку «ОК» в этом окне и возвращаемся к окну рисунка 5, но уже с заполненным тестом запроса, как представлено на рисунке 8. В окне выбираем кнопку «Далее».

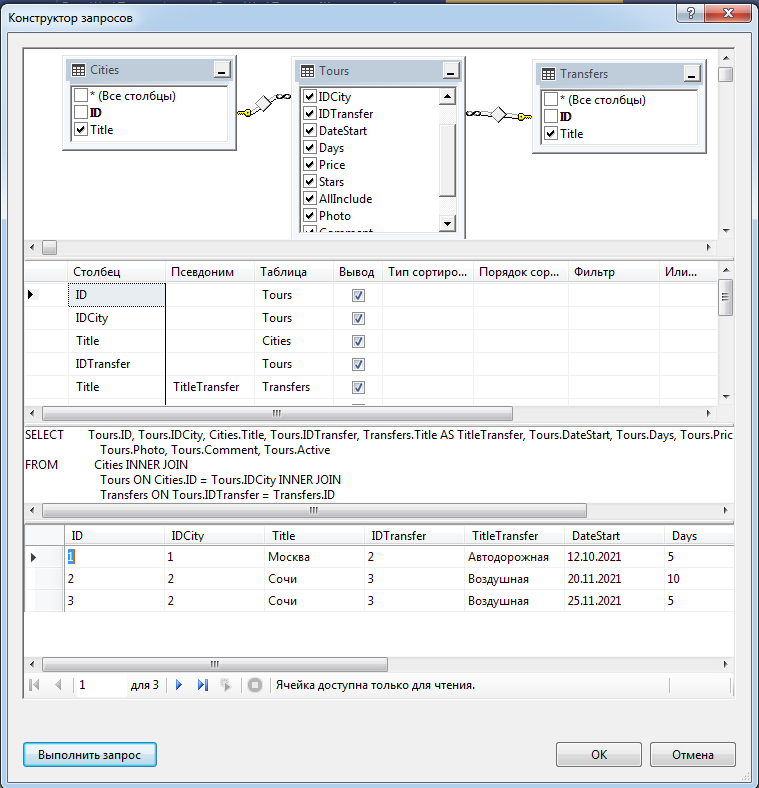


Рисунок 7 – Настроенное окно Конструктора запросов

Переходим к окну для задания имени методов адаптера, как представлено на рисунке 9. Первый метод соответствует сигнатуре метода Fill, второй — методу GetData. Имя метода по умолчанию Fill и GetData по умолчанию. Имя можно заменить на более походящее или оставить без изменения. Можно создать только один метод, сняв флажок у необязательного метода. Выбираем кнопку «Далее».

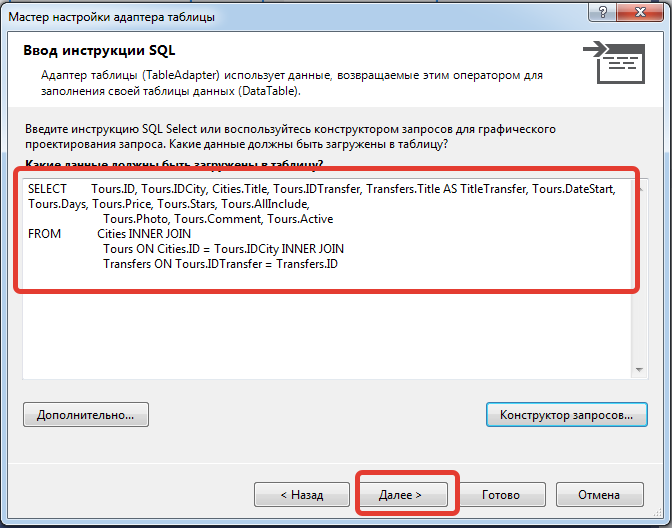


Рисунок 8 – Окно с текстом запроса

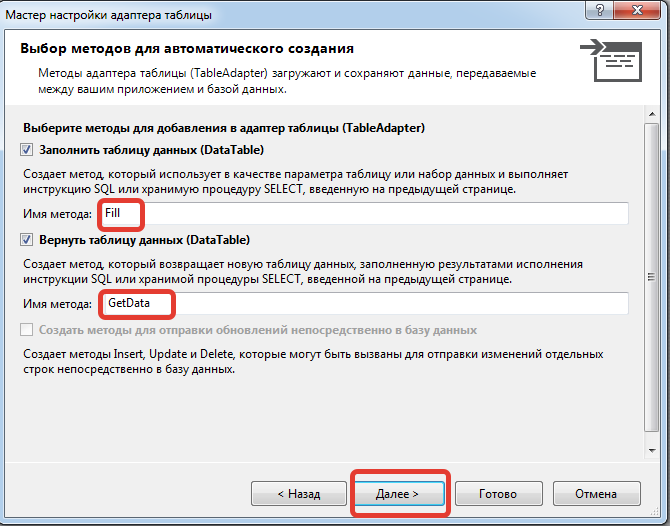


Рисунок 9– Окно для выбора имен создаваемых методов

Появляется последнее окно (см. рис 10), в котором достаточно выбрать кнопку «Готово».

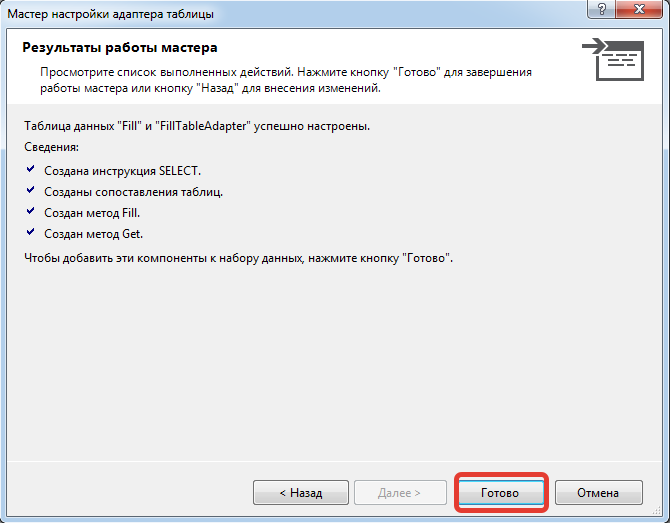


Рисунок 10– Окно окончания действий по созданию адаптера

В результате данных действий алгоритма в xsd-файла появился новый адаптер с созданными методами (см. рис.11).

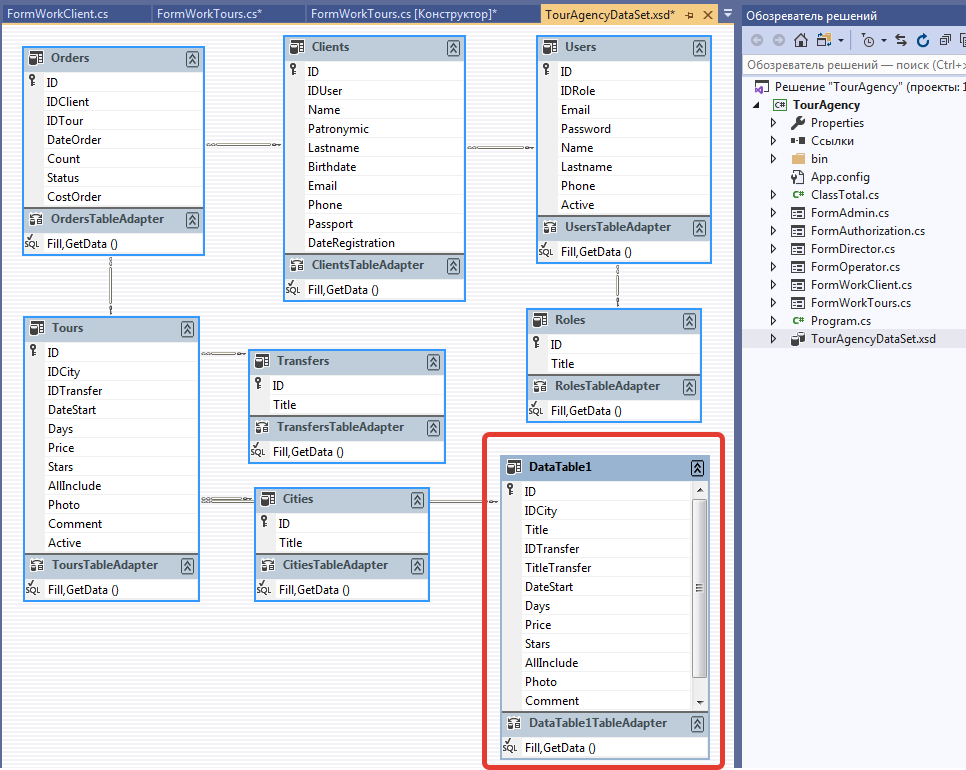


Рисунок 11– Содержимое xsd-файла с новым адаптером

Можно выполнить переименование адаптера, вызвав локальное меню к элементу адаптера, как показано на рисунке 12.

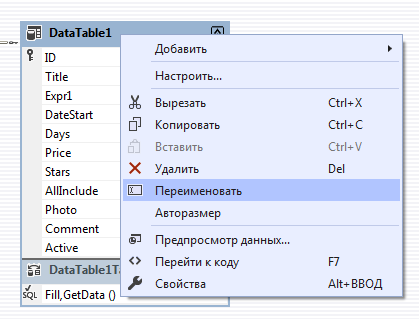


Рисунок 12 – Переименование адаптера

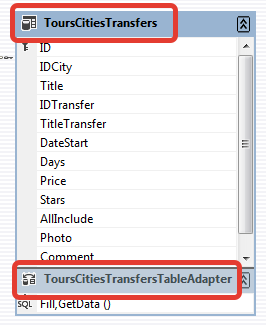
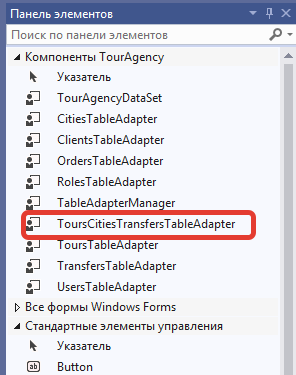
На рисунке 13 показан переименованный адаптер

Рисунок 13 – Полностью готовый адаптер



Созданный новый класс-адаптер, как компонент, должен появляться в окне «Панель элементов», как показано на рисунке 14. Если этого элемента там нет, то необходимо пересобрать приложение, и он должен появиться. Теперь можно перенести этот компонент на форму. Т.к. он невизуальный, то он будет находиться в нижней части формы, как показано на рисунке 15.

Рисунок 14 – Новый класс-компонент



Рисунок 15 – Объект созданного класса адаптера

Теперь можно писать код с применением методов созданного нового адаптера. Например, ниже представлен код, который вызывает созданный метод GetData разработанного адаптера для получения данных из смешанной таблицы и отображения их в сетке:

TourAgencyDataSet.ToursCitiesTransfersDataTable dateTours;

dateTours = this.toursCitiesTransfersTableAdapter1.GetData();

this.dataGridViewTours.DataSource = dateTours;