**Создание новых методов для адаптеров**

Адаптеры таблиц — это классы таблиц подключенной базы данных. Используется для выполнения запросов или хранимых процедур через свои методы. Адаптеры использую для осуществления основных четырех действий с таблицей БД: выбирать данные из таблиц (SQL-запрос SELECT), добавлять данные в таблицу (SQL-запрос INSERT), обновлять данные в таблице (SQL-запрос UPDATA) и удалять данные из таблицы (SQL-запрос DELETE). Другие SQL-запросы для адаптеров невыполнимы (создание БД, создание таблиц и другие).

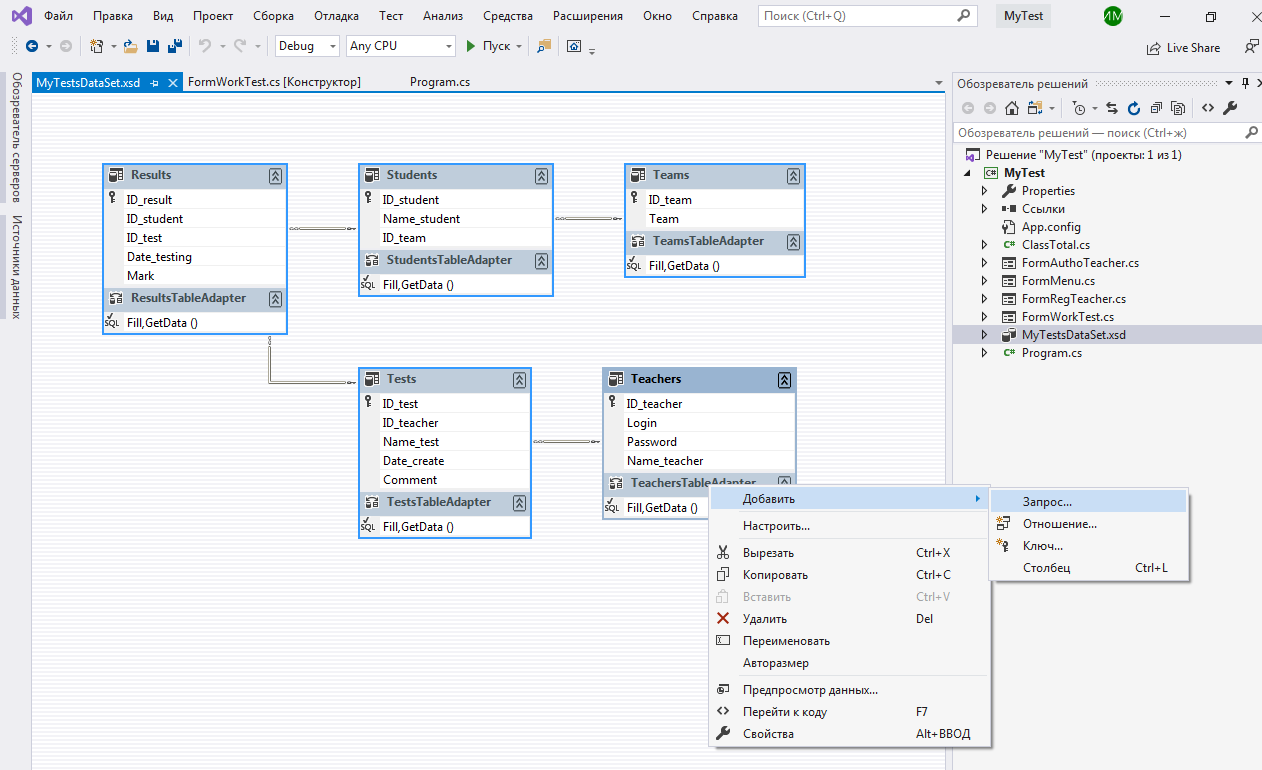
Для выполнения любого из этих четырех действий, адаптеры имеют встроенные методы. Однако, бывает необходимость создать дополнительный метод к базовому набору, который бы выполнял специальное узкое действие, например, выдавал не все записи таблицы, а удовлетворяющие некоторым условиям. Для этих целей можно создать дополнительный метод адаптера, а потом его вызывать как обычный метод к компоненту. Созданный метод адаптера напоминает технику работы с хранимыми процедурами. Отличие в том, что хранимые процедуры создаются на стороне СУБД, а для подпрограммы адаптеров – на стороне приложения.

Создание дополнительных методов адаптеров дает преимущество:

* при создании метода можно использовать Конструктор запросов, которой автоматизирует создание SQL-команды;
* метод создается один раз, а используется многократно;
* методу можно передавать параметры, которые могут участвовать при построении запроса.

Рассмотрим создание нового метода для существующего адаптера таблицы БД. Необходимо отобрать записи из таблицы, в которой значения полей «Логин» и «Пароль» совпадает с введенными пользователем значениями (фильтрация по двум полям на точное совпадение).

Двойным кликом по файлу с расширением xsd в окне «Обозреватель решений» открыть его содержимое в отдельном окне. Выбрать нужный адаптер, для которого надо создать дополнительный метод (таблицы БД, из которой надо получить сведения). На картинке – это таблица Teachers. Выбрать пункт локального меню Добавить->Запрос… (см. Рис. 1). Появится окно, представленное на рисунке 2.

Рисунок 1 – Начало создания нового метода адаптера

В этом окне надо выбрать, что создать будем именно SQL-запрос, а не хранимую процедуру. Далее появится окно, как на рисунке 3. В этом окне надо выбрать тип создаваемого SQL-запроса. Из списка нам сейчас подход первый – SQL-запрос типа SELECT, который возвращает набор записей (строк) из таблицы БД. Для других типов SQL-запросов надо будет выбирать другие варианты.

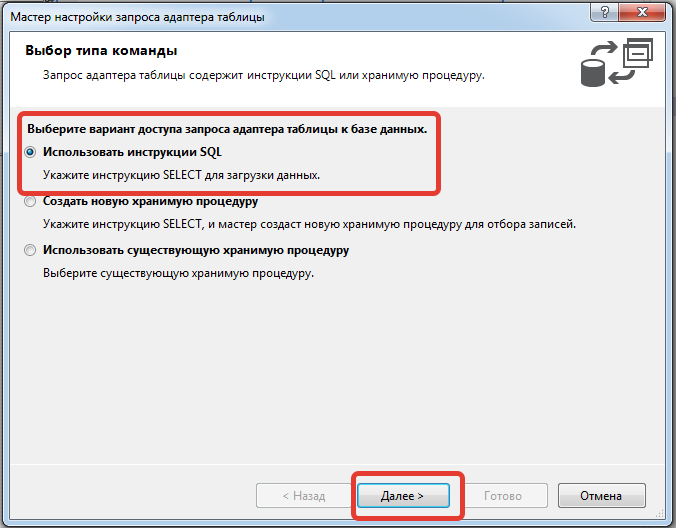


Рисунок 2 – Выбор создаваемого объекта

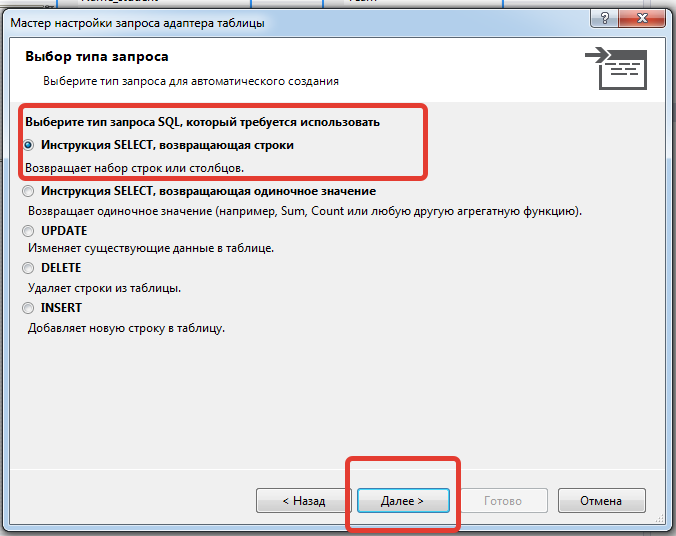


Рисунок 3 – Выбор типа SQL-запроса

После этого отобразиться окно, представленное на рисунке 4. Мастер запросов создал SQL-запрос команды SELECT на выборку всех строк таблицы. Необходимо усложнить этот запрос дополнительными условиями. Для этого можно в строке с запросом продолжить его текст, а можно воспользоваться конструктором запросов, нажав соответствующую кнопку. Появиться окно конструктора запросов, как на рисунке 4. Окно содержит четыре секции:

1 – список таблиц и полей, участвующих в построении запроса;

2 – поля, участвующие в запросе, с заданием дополнительных условий;

3 – текст сформированного запроса;

4 – пример выполнения разработанного SQL-запроса.

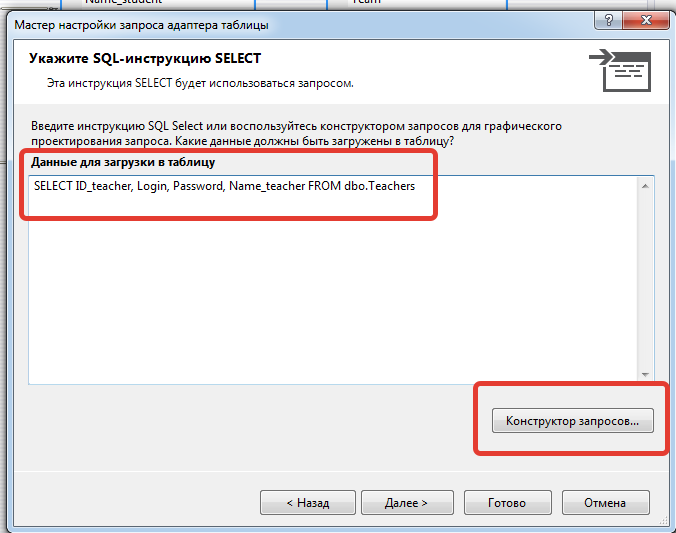


Рисунок 3 – Автоматически созданный SQL-запрос

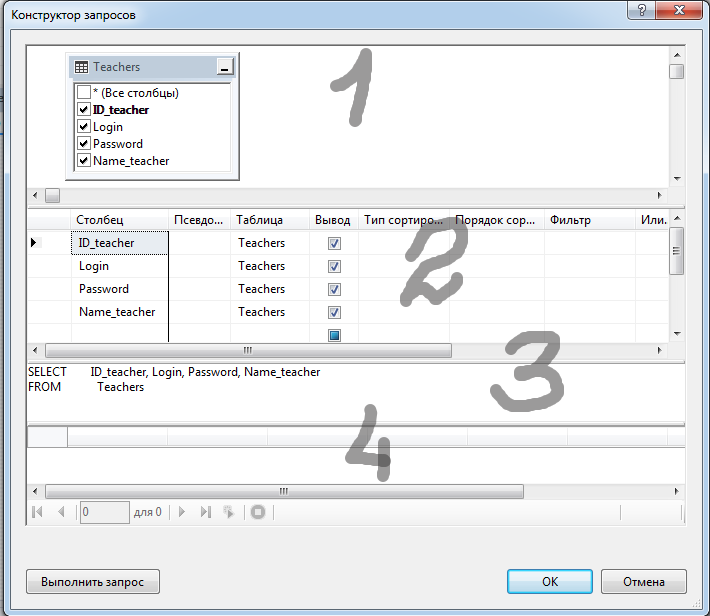


Рисунок 4 – Элементы окна конструктора запросов

Дополнительно наложим условия на необходимые поля (логин и пароль). Для этого надо перейти во вторую секция окна Конструктора и в графе «Фильтр» напротив нужных полей задать условия отбора в виде логического выражения – условие отбора записей. В условии может участвовать параметр, его имя сопровождается символом «@». Пример такого окна Конструктора запросов представлен на рисунке 5.

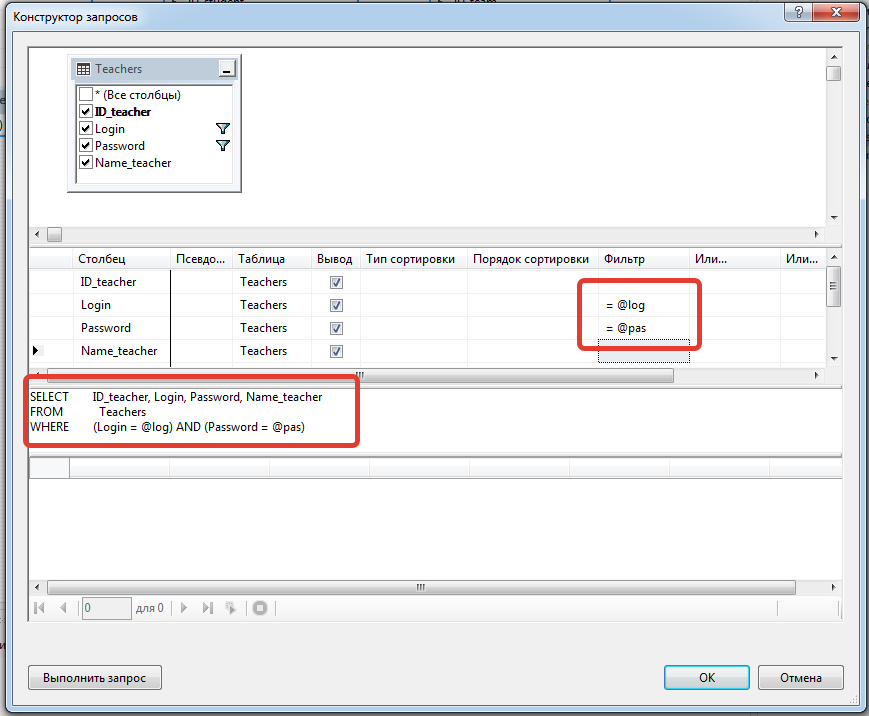


Рисунок 5 – Созданный запрос с учетом дополнительных условий на поля

Разберем заполнение окна. В 1-ой части окна указана таблица и флажками указаны поля, значения которых попадут в запрос. Во второй части окна наложены дополнительные условия по два поля: Login и Password. Значение полей должны совпадать на равенство со значениями параметров log и pas. Параметры должны сопровождаться символом @. Это означает, что их значения в запрос попадут из вне, т.е. являются входными данными для запроса. В третьем окне автоматически строится запрос с учетом выбранных параметров.

Нажмем кнопку «Выполнить запрос» для проверки работоспособности созданного запроса. Сначала появится окно с заданием конкретных значений параметров (см. рис. 6 синее выделение). После нажатия кнопки «OK» отобразить результат выполненного запроса (см. рис. 6 красное выделение). Если записей, подходящих под условие не обнаружится в БД, то окно будет пустое.

После проверки правильности созданного запроса, следует его сохранить в методе. Для этого жмем кнопку «OK» и возвращается к окну Мастера настройки запроса (см. выше рис.3). Теперь в нем отображен правильный проверенный запрос (см. рис. 7). Нажимаем кнопку «Далее» и попадаем в окно сохранения созданного запроса в методе. Для этого надо придумать осмысленное имя метода и записать его в соответствующую строку. Нажать кнопку «Далее» (см. Рис. 8).

Появится заключительное окно (см. Рис. 9), подтверждающее создание нового метода с заданной инструкцией SQL. На рисунке 10 отображено содержимое вкладки файла xsd. Видно, что адаптер, связанный с таблицей Teachers, имеет новый метод с заданным названием. Подведя указатель мыши, можно увидеть во всплывающей подсказке текст SQL-запроса этого метода (см. рис.10).

Теперь созданным методом можно пользоваться в программном кода произвольное количество раз там, где необходимо совершить фильтрацию записей в таблице за заданным двум полям.

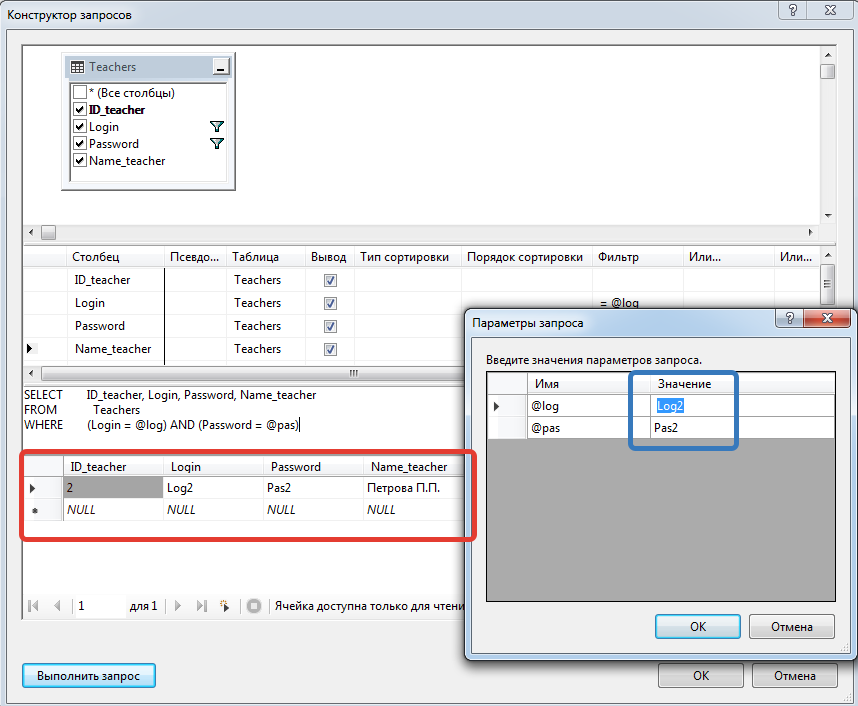


Рисунок 6 – Окна с заданными параметрами и результатами запроса

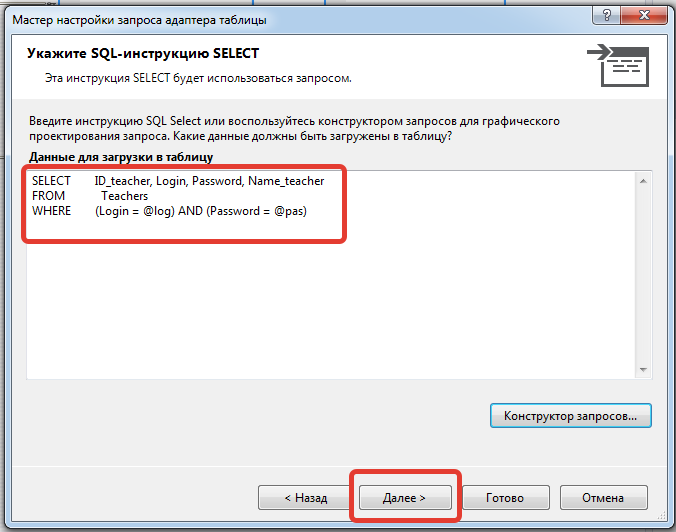


Рисунок 7 – Окно с созданным запросом

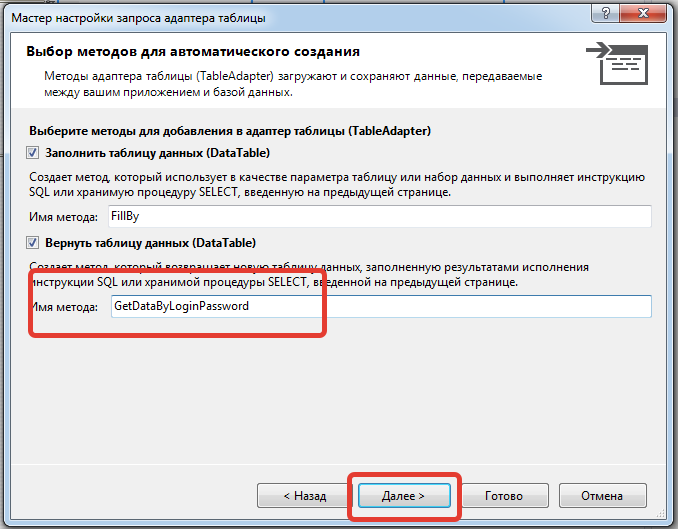


Рисунок 8 – Выбор имени метода для запроса

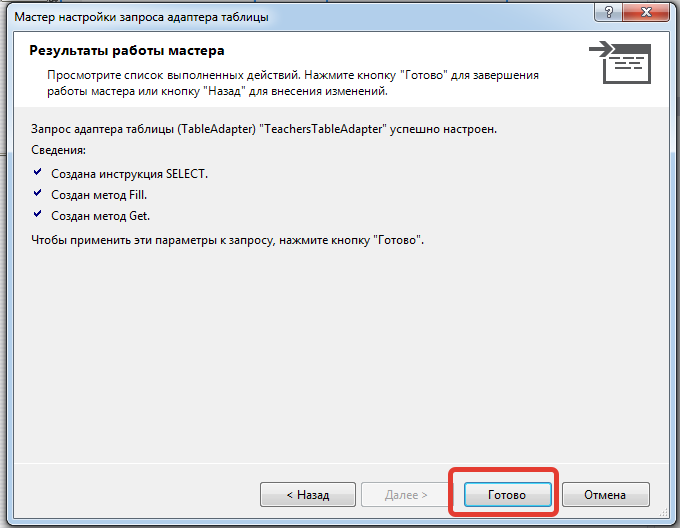


Рисунок 9 – Завершающее окно

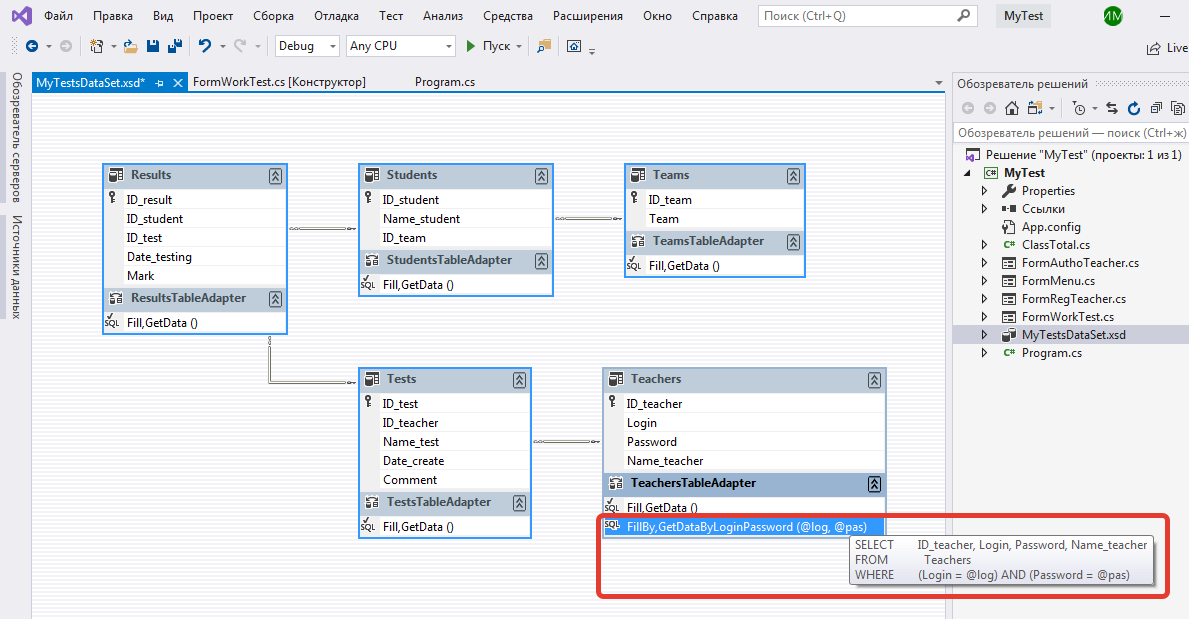


Рисунок 10 – Созданный метод

После создания нового метода, его можно использовать в программном коде, где нужно получить набор записей из таблицы БД. Метод возвращает отфильтрованный набор записей, подходящих под условие отбора. Пример использования созданного метода:

var rec = this.teachersTableAdapter1.GetDataByLoginPassword(log, pas);

if (rec.Count == 0) //Нет подходящих записей

MessageBox.Show("Таких данных нет.");

else

{

//Получение данных из вернувшихся записей

idTeacher = rec.ElementAt(0).ID\_teacher; //Первая запись и поле ID

nameTeacher = rec.ElementAt(0).Name\_teacher;

}