**Работа с записями таблицы БД через адаптеры**

1. *Получение всех записей из одной таблицы*

Для этого можно применить метод-функцию **GetData()** к объекту, созданного на основе класса адаптера нужной таблицы. Метод GetData() возвращает набор всех записей той таблицы, которая связана с адаптером, для которого вызывается метод. Метод возвращает сложный тип данных, поэтому для упрощения можно выбрать тип var (приводимый тип). Для типа var компилятор определяет и назначает наиболее подходящий тип. Может использоваться только для локальной переменной метода.

В общем виде получить все записи из таблицы в переменную records:

**var records = Объект-адаптер.GetData();**

Например: У формы есть компонент-адаптер usersTableAdapter.

var records = this.usersTableAdapter.GetData();

1. *Отображение записей таблицы БД в сетке (компонент DataGridView)*

Компонент имеет свойство DataSource, в которое записываются записи, которые нужно отобразить в компоненте.

**this.dataGridViewAuto.DataSource = records;**

1. *Получить количество записей*

**int total\_count = records.Count;**

MessageBox.Show("Всего записей в таблице БД = " + total\_count);

1. *Доступ к отдельным элементам записей набора*

Предположим, в таблицы БД есть поле с именем ID\_user, стоящим первым в списке полей.

//Получить данные конкретного столбца

DataColumn column = records.Columns["ID\_user "]; //По названию

DataColumn column = records.Columns[0]; //По его индексу в таблице

DataColumn column = records.ID\_userColumn; //По свойству

//Получить данные конкретной строки

DataRow row = records.Rows[0]; //Только по индексу строки

//Получить данные ячейки таблицы (поля) конкретной записи

int id = (int) row.ItemArray[0]; //Через строку

int id = (int) records[0].ItemArray[0]; //Через таблицу

**int id = (int) records.ElementAt(0).ID\_user; //Наиболее часто используемая**

int id = (int) records.ElementAt(0).ItemArray[0];

1. *Переходы к записям*

//Доступ к первой и последней записям набора

int id\_first = (int) records.First().ItemArray[0];

int id\_last = (int) records.Last().ID\_user;

1. *Фильтрация данных*

Часто возникает ситуация, когда нужно наложить дополнительные условия на полученный набор записей. Можно воспользоваться методом Where(), который определяет фильтр для набора, к которому вызывается. Параметром метода должно быть логическое выражение, оформленное в виде лямбда-выражения. Для каждого элемента выборки метод проверяет логическое выражение. Если оно имеет значение true, то этот элемент выборки помещается в выходную последовательность. Общий вид метода Where:

**Where(элемент\_выборки => логические выражение с элементом\_выборки)**

Метод Where возвращает сложный тип данных, поэтому можно воспользоваться типом var.

var filter = records.Where(условие фильтра);

var filter = records.Where(p => p.Login == log && p.Password == pas);

Разберем логическое лямбда-выражение в скобках метода Where. Параметр p (может быть любое имя) — это входной элемент набора, который надо проверить. Символ => означает, что параметр поступает на вход. Далее следует обычное логическое выражение, в котором может участвовать переданный параметр. В данном примере проверяется на равенство его поле Login со значением переменной log и поля Password на равенство с переменной pas. Если логическое выражение получает значение true, то данный элемент выборки попадает в результат работы метода Where.

Можно использовать фильтра без промежуточной величины

var filter = usersTableAdapter.GetData().Where(p => p.Login == log && p.Password == pas);

1. *Действия с отфильтрованными данными*
   1. Отобразить результаты в сетке. Требуется преобразование отфильтрованный данных к типу таблица с помощью метода CopyToDataTable:

this.dataGridViewStat.DataSource = filter.CopyToDataTable();

* 1. Количество записей в фильтре

int count = filter.Count();

MessageBox.Show("Количество записей, подходящих под условие отбора = "+count);

* 1. Доступ к значению ячейки. В общем виде:

filter.ElementAt(Индекс записи).имяполя;

filter.ElementAt(Индекс записи).ItemArray[Индекс поля];

Например:

int id = filter.ElementAt(0).ID\_user;

int id = (int)filter.ElementAt(0).ItemArray[0];

* 1. Доступ к первой (последней) записи фильтра

int id\_first = (int)filter.First().ItemArray[0];

int id\_last = (int)filter.Last().ItemArray[0];

* 1. Наложение дополнительного ограничения на имеющийся фильтр

EnumerableRowCollection filter\_add = filter.Where(дополнительное условие фильтра);

var filter\_add = filter.Where(дополнительное условие фильтра);

var filter\_add = filter.Where(p => p.ID\_user > 10);