

## Практическая работа №18

### Получение статистических данных

#### 1 Цель работы

1.1 Научиться выполнять сбор статистических данных используя связанные записи и агрегатные функции в LINQ-запросах.

#### 2 Литература

2.1 Фленов, М. Е. Библия C#. – 3 изд. – Санкт-Петербург: БХВ-Петербург, 2016. – URL: <https://ibooks.ru/bookshelf/353561/reading>. – Режим доступа: для зарегистрированных пользователей. – Текст : электронный. – п.8.9.

#### 3 Задание

Создать оконное приложение, состоящее из нескольких форм, на каждой должны отображаться указанные в соответствующем задании данные.

К каждой форме должен быть реализован переход с главной формы.

Для получения данных из БД использовать Entity Framework.

Данные отображать в DataGridView с запретом на вставку, удаление и редактирование.

##### 3.1 Получение минимального и максимального значения

3.1.1 Добавить в приложение форму, отображающую список товаров в табличном виде.

3.1.2 Добавить на форму два NumericUpDown для указания минимальной и максимальной цен товаров. При изменении значений цен в NumericUpDown выполнять фильтрацию данных.

3.1.3 У NumericUpDown при загрузке формы указать минимальное и максимальное допустимые значения на основе минимального и максимального значений цены из таблицы Товары, указав требуемый столбец в методах Min() и Max().

##### 3.2 Подсчет количества связанных записей (свойство Count)

3.2.1 Добавить в приложение форму, отображающую список производителей в табличном виде.

3.2.2 Изменить выборку, включив в нее только следующие столбцы: код производителя, название, количество товаров.

Для получения количества товаров использовать следующий синтаксис:

ГлавнаяТаблица.Select(item => new { ... , item.СвязаннаяТаблица.Столбец })

Вместо столбца указать Count – вернет количество записей:

Производители.Select(item => new { ... , item.Товары.Count })

##### 3.3 Подсчет суммы в связанных записях (связь M:M)

3.3.1 Добавить в приложение форму, отображающую список заказов в табличном виде.

3.3.2 Изменить выборку, включив в нее только следующие столбцы: код заказа, дату заказа, логин, стоимость заказа.

Для получения стоимости заказа использовать следующий синтаксис:

ГлавнаяТаблица.Select(item => new { ... ,  
item.ПромежуточнаяТаблица.Sum(вычисление стоимости 1 позиции) })

### 3.4 Подсчет количества записей и среднего значения в группе

3.4.1 Добавить в приложение форму, отображающую список товаров в табличном виде.

3.4.2 Изменить выборку, включив в нее только следующие столбцы: тип товара, количество товаров определенного типа, средняя цена товаров определенного типа.

Для получения количества использовать группировку по типу, метод Count() для подсчета количества элементов в группе, метод Average() для подсчета среднего значения элементов в группе.

### 3.5 Агрегация строковых данных

3.5.1 Добавить в приложение форму, отображающую список заказов в табличном виде.

3.5.2 Изменить выборку, включив в нее только следующие столбцы: код заказа, дату заказа, названия товаров.

Для получения названий товаров использовать поле Товар из таблицы Состав:

Столбец = item.**СвязаннаяТаблица.Select**(выборка требуемого столбца)

Полученную выборку с номерами заказов, датами и названиями товаров привести к списку.

3.5.3 На основе выборки из п.3.5.2 создать новую выборку, объединяющую строковые данные в одну строку. Для объединения к списку названий применить метод Aggregate():

item.**Список.Aggregate**((a, b) => (a + ", " + b))

3.5.4 Добавить проверку на то, что в списке есть элементы. Если их нет, присваивать пустую строку.

## 4 Порядок выполнения работы

4.1 Выполнить все задания из п.3 в одном проекте PractWork18.

4.2 Ответить на контрольные вопросы.

## 5 Содержание отчета

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

## 6 Контрольные вопросы

6.1 Что возвращают агрегатные функции Sum(), Min() и Max()?

6.2 Как задать условие в методе Count()?

6.3 Для чего используется метод Aggregate()?

6.4 Как указать столбцы в методах Sum(), Min() и Max()?

6.5 Для чего применяется метод GroupBy()?