

# EAGER LOADING C#

(ДОКЛАД ПО ДИСЦИПЛИНЕ «ПРОЕКТНЫЙ  
ПРАКТИКУМ **2**»)

ВЫПОЛНИЛА

СТУДЕНТКА ГРУППЫ **Б07-021-1**

СЕМЕРИКОВА МАРИЯ

# ЗАГРУЗКА СВЯЗАННЫХ ДАННЫХ

**В Entity Framework существует три подхода для загрузки связанных данных:**

- **«отложенная загрузка» (lazy loading);**
- **«прямая загрузка» (eager loading);**
- **«явная загрузка» (explicit loading).**

# EAGER LOADING

**В этом примере базе данных будет отправляться всего один запрос при инициализации коллекции customers.**

```
C#  
  
public static void EagerLoading()  
{  
    SampleContext context = new SampleContext();  
  
    context.Database.Log = (s => System.Diagnostics.Debug.WriteLine(s));  
  
    // Загрузить всех покупателей и связанные с ними заказы  
    List<Customer> customers = context.Customers  
        .Include(c => c.Orders)  
        .ToList();    // +1 запрос к базе  
  
    // ... какой-то код работы с данными покупателей  
  
    // Получить все их заказы  
    List<Order> orders = customers.SelectMany(c => c.Orders)  
        // Запрос к базе данных не выполняется,  
        // т.к. данные уже были извлечены  
        // ранее с помощью прямой загрузки  
        .ToList();  
}
```

Рисунок 1 - Метод EagerLoading

# EAGER LOADING

```
C#  
  
public class Customer  
{  
    // ...  
  
    public Profile Profile { get; set; }  
}  
  
public class Order  
{  
    // ...  
  
    public List<OrderLines> Lines { get; set; }  
}  
  
public class Profile  
{  
    [Key]  
    [ForeignKey("CustomerOf")]  
    public int CustomerId { get; set; }  
    public DateTime RegistrationDate { get; set; }  
    public DateTime LastLoginDate { get; set; }  
  
    public Customer CustomerOf { get; set; }  
}  
  
public class OrderLines  
{  
    public int OrderLinesId { get; set; }  
    public string Address { get; set; }  
}
```

**Класс Profile  
связан с классом  
Customer  
связью один-к-  
одному, а класс  
OrderLine связан  
с классом Order  
связью один-ко-  
многим.**

Рисунок 2 – Добавление классов

# EAGER LOADING

```
C#  
  
context.Customers  
    .Include(c => c.Profile)  
    .Include(c => c.Orders.Select(o => o.Lines))  
    .ToList();
```

Рисунок 3 – Загрузка связанных данных

**Для загрузки связанных данных из таблицы Profiles нужно будет использовать отдельный вызов метода Include(), а для загрузки связанных данных из OrderLines нужно указать метод Select() со ссылкой на этот класс, при вызове метод Include() для Orders (т.к. OrderLines напрямую не связан с Customer).**

# EAGER LOADING

```
C#  
  
Customer customer = context.Customers  
    .Include(c => c.Profile)  
    .Include(c => c.Orders.Select(o => o.Lines))  
    .Where(c => c.CustomerId == 2)  
    .FirstOrDefault();  
  
if (customer.Orders != null)  
    foreach (Order order in customer.Orders)  
        Console.WriteLine(order.ProductName);
```

Рисунок 4 - Запрос

**Запрос извлекает все связанные данные для покупателя с идентификатором равным 2.**

# EAGER LOADING

Прямая загрузка приводит к возврату всех связанных сущностей вместе с запрошенными сущностями в единственном запросе. Это означает, что, несмотря на установку только одного соединения с источником данных, в начальном запросе происходит возврат большего объема данных. Кроме того, применение путей запроса приводят к получению более сложного запроса из-за дополнительных соединений, которые требуются в запросе к источнику данных.