

Содержание

Сумма по справочникам выбранных объектов	3
Просроченные задачи пользователей	3
Задачи за 3 дня до начала (Задать дату начала)	3
Задача с датой окончания и статус отчёта по задаче	4
Контрагенты, у которых есть не закончившиеся инсталляции (через связи)	4
Прошлогодние задачи конкретного пользователя	5
Сделки созданные за апрель 2019	5
Получение ID родителя	6
Поиск родителя	6

Примеры LINQ-запросов

Сумма по справочникам выбранных объектов

```
var deals = dataContext.Kontragent_9e5de8_List
    .Where(z => z.Name.Contains("Фармент"))
    .Select(z => new {z.Name, Trudochasy =
z.GetChildrenHierarchy<Uchet_vremeni_3156dc>().Sum(r => r.Chasi), Dengi =
z.GetChildrenHierarchy<Oplata_45b34c>().Sum(r => r.Summa)})
    .OrderBy(z=> z.Name);

return deals;
```

Просроченные задачи пользователей

```
var dToday = DateTime.Today;

var tasks = dataContext.Zadacha_c88ec2_List
    .Where(z => z.SystemEndDate < dToday && (z.Status ==
ProjectStatus.InWork || z.Status == ProjectStatus.NotStarted) )
    .OrderBy(z => z.Responsible.FirstName).ThenBy(z =>
z.Responsible.LastName).ThenBy(z => z.Name)
    .Select(z => new { z.Name,
                    user_id = z.Responsible.Id,
                    Ispolnitel = String.Concat(z.Responsible.FirstName, "
", z.Responsible.LastName),
                    z.Responsible.Email,
                    z.SystemStartDate, z.SystemEndDate
                }
            );

return tasks;
```

Задачи за 3 дня до начала (Задать дату начала)

```
var dStart = new DateTime(2019, 5, 1);
var dEnd = dStart.AddDays(4);

var deals = dataContext.SMART__zadacha_45a3a0_List
    .Where(z => z.PlannedStartDate >= dStart && z.PlannedStartDate < dEnd )
    .OrderBy(z => z.Name)
    .Select(z => new { z.Name,
                    boss = String.Concat(z.Owner.LastName, " ",
z.Owner.FirstName),
```

```

        responsible = String.Concat(z.Responsible.LastName, "
", z.Responsible.FirstName),
        Creator = String.Concat(z.Creator.LastName, " ",
z.Creator.FirstName),
        Status = (z.Status == ProjectStatus.InWork ? "В
работе" :
                (z.Status == ProjectStatus.Complete ?
"Выполнен" :
                (z.Status == ProjectStatus.Cancelled ?
"Отменен" :
                (z.Status == ProjectStatus.Freeze ?
"Отложен" :
                (z.Status == ProjectStatus.NotStarted
? "Не начат" :
                "Готов к проверке"
                )
                )
                )
                )
                ),
        z.PlannedStartDate, z.SystemStartDate,
z.ActualEndDate, dStart, dEnd});
return deals;

```

Задача с датой окончания и статус отчёта по задаче

```

var tasks = dataContext.Zadacha_c88ec2_List
    .Where(z =>
z.GetChildrenHierarchy<Otchet_o_statuse_zadachi_3c7214>().Any())
    .OrderByDescending(z => z.SystemEndDate)
    .Select(z => new
    {
        z.Name,
        z.PlannedEndDate, //Плановая ДО (ограничение)
        z.SystemEndDate, //Расчетная ДО (из планировщика)
        Otchet_o_statuse =
z.GetChildrenHierarchy<Otchet_o_statuse_zadachi_3c7214>()
            .OrderByDescending(r => r.Date)
            .FirstOrDefault()
            .Status_056648.Name
    });

return tasks;

```

Контрагенты, у которых есть не закончившиеся

инсталляции (через связи)

```
var dStart = DateTime.Now;

var Kontragents = dataContext.Kontragent_9e5de8_List
    .Where(z => z.Installyaciya_180ce0.Where(p => p.Data_okonchaniya_licenzii
    >= dStart).Any() )
    .OrderBy(z => z.Name)
    .Select(z => new {
        z.Name,
        z.Id,
        ResposibleId = z.Responsible.Id,
        z.Installyaciya_180ce0.FirstOrDefault().Data_okonchaniya_licenzii,
        Nazvanie_inst = z.Installyaciya_180ce0.FirstOrDefault().Name,
        Chislo = z.Installyaciya_180ce0.Count()
    });

return Kontragents;
```

Прошлогодние задачи конкретного пользователя

```
var personId = parameters.GetValueOrDefault<Guid>("PersonId", new
Guid("50330e78-01c7-4280-9c74-2be072991628"));
var previousYear = DateTime.Now.AddYears(-1).Year;

var tasks = dataContext.Zadacha_razrabotki_6ad838_List
    .Where(z => z.Celj_Opisanie != null &&
        z.PlannedStartDate.Value.Year == previousYear &&
        z.Responsible.Id == personId)
    .OrderByDescending(z => z.PlannedStartDate)
    .Select(z => new { z.Name, z.SystemStartDate, z.SystemEndDate,
z.Celj_Opisanie, z.Trudoyomkostj });

return tasks;
```

Сделки созданные за апрель 2019

```
var deals = dataContext.Sdelka_3f169d_List
    .Where(z => z.CreationDate >= new DateTime(2019, 4, 1) && z.CreationDate
< new DateTime(2019, 5, 1) )
    .OrderBy(z => z.Name)
    .Select(z => new { Sdelka = z.Name,
        KontragentName =
z.GetParentHierarchy<Kontragent_9e5de8>().FirstOrDefault().Name,
        boss = String.Concat(z.Owner.LastName, " ",
z.Owner.FirstName),
```

```

        responsible = String.Concat(z.Responsible.LastName, "
", z.Responsible.FirstName),
        Creator = String.Concat(z.Creator.LastName, " ",
z.Creator.FirstName),
        Status = (z.Status == ProjectStatus.InWork ? "В
работе" :
                (z.Status == ProjectStatus.Complete ?
"Выполнен" :
                (z.Status == ProjectStatus.Cancelled ?
"Отменен" :
                (z.Status == ProjectStatus.Freeze ?
"Отложен" :
                (z.Status == ProjectStatus.NotStarted
? "Не начат" :
                "Готов к проверке"
                )
                )
                )
                )
                ),
        Prichina_otkaza = z.Prichini_otkaza_5ad9a4.Name,
        z.SystemEndDate, z.SystemStartDate, z.ActualEndDate,
        delegirovano_menedjeru = ((DateTime?)
z.Data_delegirovaniya_sotrudniku).HasValue ? 1 : 0 ,
        Sdelka_length =
DbFunctions.DiffDays(z.SystemStartDate, z.SystemEndDate) });
return deals;

```

Получение ID родителя

у объекта есть метод `public TCustomType GetParent<TCustomType>()` - получение родителя, если родитель не `TCustomType`, то вернется `null` - возвращает прямого (ровно на 1 уровень вверх) родителя

если надо выше, то - `public ICollection<TCustomType> GetParentHierarchy<TCustomType>(bool includeRoot)` получение родителей с учетом иерархии

Поиск родителя

```

var nodeGUID =
parameters.GetValueOrDefault<Guid?>(QueryParameters.SpreadsheetReport.ProjectId, new Guid("3e637a10-51b8-4a02-89fe-35b645969ca5"));

var projects = DataContext.Projects
    .Where(p => p.GetParentHierarchy<Proekt_vnedreniya_15f54a>(true).Any(i => i.Id == nodeGUID))
    .OrderBy(p => p.Parent.Id)

```

```
.OrderBy(p => p.CreationDate)
.Select(p => new{
    p.Name,
    object_id = p.Id,
    ParentId = p.Parent.Id
});
return projects;
```

From:

<https://wiki.a2nta.ru/> - Wiki [3.x]

Permanent link:

<https://wiki.a2nta.ru/doku.php/product/linq/exaples?rev=1755593934>

Last update: **19.08.2025 08:58**

