

## Содержание

<b>Сумма по справочникам выбранных объектов</b> .....	3
<b>Просроченные задачи пользователей</b> .....	3
<b>Задачи за 3 дня до начала (Задать дату начала)</b> .....	3
<b>Задача с датой окончания и статус отчёта по задаче</b> .....	4
<b>Контрагенты, у которых есть не закончившиеся инсталляции (через связи)</b> .....	4
<b>Прошлогодние задачи конкретного пользователя</b> .....	5
<b>Сделки созданные за апрель 2019</b> .....	5
<b>Получение ID родителя</b> .....	6
Поиск родителя .....	6
<b>Запрос по проекту (с выводом количества разного рода объектов) для анализа состояния проекта</b> .....	7
<b>Объединение плановых и фактических трудозатрат</b> .....	12



# 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_stature_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_stature =
z.GetChildrenHierarchy<Otchet_o_stature_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 = dbContext.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 = dbContext.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 = dbContext.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;
```

## Запрос по проекту (с выводом количества разного рода объектов) для анализа состояния проекта

Запрос по проекту, позволяющий посчитать количество объектов для анализа состояния проекта:

1. Все объекты.
2. Все конечные объекты.
3. Количество работ без зависимостей.
4. Количество работ с фиксированными сроками.
5. Всего объектов просрочено.
6. Количество конечных работ с несвоевременным началом.
7. Количество приоритетных задач.
8. Конечных работе без отчета о ходе работ и т.п.

```
var id =
parameters.GetValueOrDefault<Guid?>(QueryParameters.SpreadsheetReport.ProjectId, null /* new Guid("273bf10a-f403-4e62-a95d-464ad94616af")*/ );
var idProject = id;
var сегодня = DateTime.Now;
var сегодня_7дн = сегодня.AddDays (-7);
var сегодня_14дн = сегодня.AddDays (-14);
var сегодня_плюс_12 = сегодня.AddMonths(12);
var сегодня_плюс_3 = сегодня.AddMonths(3);
var сегодня_мин_3_мес = сегодня.AddMonths(-3);
// var currentYear = new DateTime(сегодня.Year.Month,1);
var defaultCalendar = workCalendars.GetDefaultCalendar();

var dToday = DateTime.Now;
var child = dataContext.Projects
    .Where(p => p.GetParentHierarchy<Project>(false).Any(a => a.Id == idProject))
    .Where(a => a.GetChildren<Project>().Any() == false )
    .Where(a => a.BaselinePlanEndDate != null && (a.Status == ProjectStatus.NotStarted || a.Status == ProjectStatus.InWork))
    .Select(p => new {
        p.Id,
        ParentId = p.Parent.Id,
        Calendar = p.CalendarId,
        Start = p.SystemStartDate,
        End = p.SystemEndDate,
```

```

        UDN = p.BaselinePlanStartDate,
        UDO = p.BaselinePlanEndDate,
        Status1 = p.Status,
//      HasChild = p.GetChildren<Project>().Any() == false
    })
    .ToList()
    .Where(a =>
(Math.Round(workCalendars.GetWorkCalendar(a.Calendar).GetWorktimeDays(a.Start.Value.Date, a.End.Value.Date)) < (a.UDO != null ?
Math.Round(workCalendars.GetWorkCalendar(a.Calendar).GetWorktimeDays(a.UDN.Value.Date, a.UDO.Value.Date)) : 0)))
    .Select(p => new
    {
        p.Id,
        p.ParentId,
        p.Calendar,
        p.Start,
        p.End,
        p.UDN,
        p.UDO,
        p.Status1,
        План_длительность =
Math.Round(workCalendars.GetWorkCalendar(p.Calendar).GetWorktimeDays(p.Start.Value.Date, p.End.Value.Date)),
        УТВ_длительность = p.UDO != null ?
Math.Round(workCalendars.GetWorkCalendar(p.Calendar).GetWorktimeDays(p.UDN.Value.Date, p.UDO.Value.Date)) : 0,
    });

var projects = dataContext.Projects
    .Where(p => p.GetParentHierarchy<Project>(true).Any(r => r.Id == idProject))
    .Where(p => p is Proekt_tipovoj_b9f9ae || p is Proekt_tipovoj_2_82b061 || p is Proekt_Lean_a65539)
    .Where(p => p.Name != null)

    .Select(p => new {
        ID_проекта = p.Id,
        Название_проекта = p.Name,
        Руководитель = String.Concat(p.Owner.LastName, " ", p.Owner.FirstName),
        План_завершения = DbFunctions.TruncateTime(p.SystemEndDate),
        SystemStartDate = DbFunctions.TruncateTime(p.SystemStartDate),
        SystemEndDate = DbFunctions.TruncateTime(p.SystemEndDate),
        Утвержд_завершение = DbFunctions.TruncateTime(p.BaselinePlanEndDate),
        Утвержд_начало = DbFunctions.TruncateTime(p.BaselinePlanStartDate),
        Процент_завершения = p.PercentComplete,
        Всего_объектов = p.GetChildrenHierarchy<Project>(false).Select(a => a.Name != null).Count(),
        Всего_конечных_объектов = p.GetChildrenHierarchy<Project>(false).Where(a => a.GetChildren<Project>().Any() == false).Count(),
        Конечных_год_вперед_1 = p.GetChildrenHierarchy<Project>(false).Where(a => a.GetChildren<Project>().Any() == false && a.SystemEndDate >= сегодня &&

```

```

а.SystemEndDate <= сегодня_плюс_12).Count(),
    Конечных_более1мес_1 = p.GetChildrenHierarchy<Project>(false).Where(a =>
а.GetChildren<Project>().Any() == false && а.SystemEndDate >= сегодня &&
а.SystemEndDate <= сегодня_плюс_12 &&
DbFunctions.DiffDays(а.SystemStartDate,а.SystemEndDate)> 31).Count(),
    Колво_работ_без_связей_2 = p.GetChildrenHierarchy<Project>(false).Where(а =>
а.GetChildren<Project>().Any() == false && а.Dependencies.In.Count ==
0).Count(),
    Колво_работ_с_фиксом_3 = p.GetChildrenHierarchy<Project>(false).Where(а =>
а.GetChildren<Project>().Any() == false && а.PlannedStartDate.HasValue ||
а.PlannedEndDate.HasValue).Count(),
    Всего_объектов_просрочено_5 = p.GetChildrenHierarchy<Project>(false).Where(а
=> а.Status == ProjectStatus.NotStarted || а.Status == ProjectStatus.InWork
|| а.Status == ProjectStatus.Ready).Where(s =>
DbFunctions.TruncateTime(s.SystemEndDate) <
DbFunctions.TruncateTime(сегодня)).Count(),
    Всего_объектов_критич_6 = p.GetChildrenHierarchy<Project>(false).Where(а =>
а.Priority == 1).Count(),

    /*В ПОСТОБРАБОТКЕ СЧИТАЕТ РАБ.ДНИ*/
    Конечных_длит_7 = p.GetChildrenHierarchy<Project>(false).Where(а =>
а.GetChildren<Project>().Any() == false && а.BaselinePlanEndDate != null &&
(а.Status == ProjectStatus.NotStarted || а.Status == ProjectStatus.InWork)
&& DbFunctions.DiffDays(а.SystemStartDate,а.SystemEndDate) <
DbFunctions.DiffDays(а.BaselinePlanStartDate,а.BaselinePlanEndDate)).Count()
,

    Длительность_проекта = DbFunctions.DiffDays(p.SystemStartDate,
p.SystemEndDate),
    Длительность_проекта_9 = (100 -
((DbFunctions.DiffDays(p.BaselinePlanEndDate,
p.BaselinePlanStartDate))/(DbFunctions.DiffDays(p.SystemEndDate,
p.SystemStartDate))) * 100),
    Конечных_несвоевр_начало_12 = p.GetChildrenHierarchy<Project>(false).Where(а
=> а.GetChildren<Project>().Any() == false && а.Priority != 1).Where (s =>
s.ActualStartDate != null ?
DbFunctions.DiffDays(s.SystemStartDate,s.ActualStartDate) > 7 :
s.SystemStartDate < сегодня_7дн).Count(),
    /*ПРОВЕРИТЬ*/
    Конечных_несвоевр_начало_критич_12 =
p.GetChildrenHierarchy<Project>(false).Where(а =>
а.GetChildren<Project>().Any() == false && а.Priority == 1).Where (s =>
s.ActualStartDate != null ?
DbFunctions.DiffDays(s.SystemStartDate,s.ActualStartDate) > 7 :
s.SystemStartDate < сегодня_7дн).Count(),

    Конечных_без_отчета_13 = p.GetChildrenHierarchy<Project>(false).Where(а =>
а.GetChildren<Project>().Any() == false && а.Priority != 1).Where(а =>
(а.Status == ProjectStatus.InWork && а.SystemStartDate < сегодня_14дн) || (
а.Status == ProjectStatus.NotStarted &&
DbFunctions.TruncateTime(а.SystemStartDate) <

```

```
DbFunctions.TruncateTime(сегодня) )).Where(a =>
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date < сегодня_14дн ||
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date == null ).Count(),

Конечных_крит_без_отчета_13 = p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority == 1).Where(a =>
(a.Status == ProjectStatus.InWork && a.SystemStartDate < сегодня_14дн) || (
a.Status == ProjectStatus.NotStarted &&
DbFunctions.TruncateTime(a.SystemStartDate) <
DbFunctions.TruncateTime(сегодня) )).Where(a =>
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date < сегодня_14дн ||
a.GetChildren<Otchet_o_hode_rabot_868b12>().OrderByDescending(r =>
r.Date).FirstOrDefault().Date == null ).Count(),

Конечных_просроченных_без_отчета_14 =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority != 1)
.Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
.Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate)).Count(),
Конечных_просроченных_14_со_статусом =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority != 1)
.Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
.Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate))
.Where(h =>
h.GetChildren<Otchet_o_hode_rabot_868b12>().Where(r => r.Date >=
сегодня_мин_3_мес).Any(s => s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Krizis_96cbc3ec_Id ||
s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Problemi_47ab098b_Id)).Count(),

Конечных_критич_без_отчета_14 =
p.GetChildrenHierarchy<Project>(false).Where(a =>
a.GetChildren<Project>().Any() == false && a.Priority == 1)
.Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
.Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate)).Count(),
```

```

    Конечных_критич_14_со_статусом =
    p.ChildrenHierarchy<Project>(false).Where(a =>
    a.Children<Project>().Any() == false && a.Priority == 1)
        .Where(a => a.Status ==
ProjectStatus.InWork || a.Status == ProjectStatus.NotStarted || a.Status ==
ProjectStatus.Freeze)
            .Where(s => s.BaselinePlanEndDate
!= null && DbFunctions.TruncateTime(сегодня) >
DbFunctions.TruncateTime(s.BaselinePlanEndDate))
                .Where(h =>
h.Children<Otchet_o_hode_rabot_868b12>().Where(r => r.Date >=
сегодня_мин_3_мес).Any(s => s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Krizis_96cbc3ec_Id ||
s.Indikator_problem_b7386a.Id ==
Classifier_Indikator_problem_6c3d8e.Problemi_47ab098b_Id)).Count(),

    new_колво_мес_KT_4 = p.ChildrenHierarchy<Project>(false).Where(k => k
is KT_1_f0f97e || k is KT_2_6393db || k is KT_3_f1cf19).Where(a =>
a.SystemEndDate >= сегодня && a.SystemEndDate <= сегодня_плюс_12).GroupBy(g =>
g.SystemEndDate.Value.Month).Count(),
    new_колво_мес_проект_4 = DbFunctions.DiffMonths(сегодня, p.SystemEndDate <=
сегодня_плюс_12 ? p.SystemEndDate : сегодня_плюс_12),
    Сегодня = DbFunctions.TruncateTime(сегодня),
    Плюс_12_мес_ = сегодня_плюс_12,
    Тек_мес_14дн = сегодня_14дн,
})
.ToList()

.Select(p => new{
    p.ID_проекта,
    p.Название_проекта,
    p.Руководитель,
    p.SystemStartDate,
    p.SystemEndDate,
    p.Утвержд_завершение,
    p.Утвержд_начало,
    p.Процент_завершения,
    p.Всего_объектов,
    p.Всего_конечных_объектов,
    p.Конечных_год_вперед_1,
    p.Конечных_более1мес_1,
    Колво_работ_без_связей_2 = p.Колво_работ_без_связей_2 == 0 ? 0 :
p.Колво_работ_без_связей_2 - 1,
    p.Колво_работ_с_фиксом_3,
    p.Всего_объектов_просрочено_5,
    p.Всего_объектов_критич_6,
    Конечных_длит_7 = child.Count(),
    p.Длительность_проекта,
    p.Длительность_проекта_9,
    p.Конечных_несвоевр_начало_12,
    p.Конечных_несвоевр_начало_критич_12,

```

```
    p.Конечных_без_отчета_13,  
    p.Конечных_крит_без_отчета_13,  
    Конечных_просроченных_без_отчета_14 = p.Конечных_просроченных_без_отчета_14 -  
p.Конечных_просроченных_14_со_статусом,  
    Конечных_критич_без_отчета_14 = p.Конечных_критич_без_отчета_14 -  
p.Конечных_критич_14_со_статусом,  
    p.new_колво_мес_КТ_4,  
    p.new_колво_мес_проект_4,  
    p.Сегодня,  
  
    })  
  
    .ToList();  
return projects;
```

## Объединение плановых и фактических трудозатрат

```
var Portfel = parameters.GetValueOrDefault<string>("Portfel",  
"14c8bd68-96c9-46e5-b870-5ef4e341e4f9");  
var Текущие_проекты = new Guid("1d1ac40d-156b-4726-b2be-877aae71dea");  
  
var plan = dataContext.ProjectCosts  
    .Where(p => p.Project.GetParentHierarchy<Katalog_743df4>(true).Any(r  
=> r.Id == Текущие_проекты))  
    .Select(c => new  
    {  
        ProjectUID =  
(Guid?)c.Project.GetParentHierarchy<Proekt_a3e2fb>(true).FirstOrDefault().Id  
,  
        TaskUID = c.ProjectId,  
        // ProjectName = c.Project.Name,  
        // ResourceUID_plan = c.PersonId,  
        // ResourceUID_fakt = (String?) "факт",  
        Date = (DateTime?)c.Date,  
        Plan = (double?)c.Value,  
        Fact = (double?)null  
    })  
    .ToList();  
  
var fact = dataContext.Fakticheskie_trudozatrati_977ff0_List  
    .Where(p => p.Project.GetParentHierarchy<Katalog_743df4>(true).Any(r  
=> r.Id == Текущие_проекты))  
    .Select(f => new  
    {  
        ProjectUID =  
(Guid?)f.Project.GetParentHierarchy<Proekt_a3e2fb>(true).FirstOrDefault().Id  
,  
        TaskUID = f.Project.Id,  
        // ProjectName = f.Project.Name,
```

```
// ResourceUID_plan = (String?) "план",
// ResourceUID_fakt = f.Resurs_caaaaa,
    f.Date,
    Plan = (double?)null,
    Fact = (double?)f.Summa_06ff93,
    })
.ToList();

var costs = plan.Union(fact)
    .GroupBy(c => (c.Date, c.TaskUID),
        (keys, values) => new
        {
            values.FirstOrDefault().ProjectUID,
            keys.TaskUID,
            // values.FirstOrDefault().ProjectName,
            keys.Date,
            Plan = values.Sum(s => s.Plan),
            Fact = values.Sum(s => s.Fact)
        })
    .ToList();

return costs;
```

From:

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

Permanent link:

<https://wiki.a2nta.ru/doku.php/product/linq/exaples?rev=1757923326>

Last update: **15.09.2025 08:02**

