

Terrasoft[®]

 ПРОГРАММНЫЕ
ТЕХНОЛОГИИ

«Центр Разработки и Внедрения
Террасофт Поволжье»

Работа с данными

Содержание

Классы доступа к БД

Select, Insert, Update, Delete

EntitySchemaQuery

Классы доступа к БД

Классы доступа к БД

Классы доступа к БД без учета прав доступа

Select

Insert

Update

Delete

InsertSelect

StoredProcedure

UserDefinedFunction

CustomQuery

С учетом прав доступа

EntitySchemaQuery

Классы доступа к БД без учета
прав доступа

Select, Insert, Update, Delete

Select

```
// Получить количество активностей по ответственным
var select = new Select(userConnection)
    .Column(Func.Count("Id"))
    .Column("OwnerId")
    .From("Activity")
    .GroupBy("OwnerId") as Select;
```

Update

```
new Update(UserConnection, "SocialAccount")
    .Set("Public", Column.Parameter(false))
    .Where("Id").IsEqual(Column.Parameter(NewSocialAccountId))
    .Execute();
```

StoredProcedure, UserDefinedFunction

StoredProcedure

```
StoredProcedure setRecordPositionProcedure =  
    new StoredProcedure(UserConnection, "tsp_ChangePrimaryCulture")  
        .WithParameter("SysCultureId", item.Key.ToString()) as StoredProcedure;  
setRecordPositionProcedure.Execute();
```

UserDefinedFunction

```
var userDefinedFunction = new UserDefinedFunction(userConnection, "fn_IsGuid")  
    .WithParameter("ObjectId", checkId);
```

InsertSelect

```
var insertSelect = new InsertSelect(UserConnection).
    Into("SysUserInRole").
    Set("CreatedOn", "ModifiedOn", "SysRoleId", "SysUserId").
    FromSelect(new Select(UserConnection)
        .Column(Column.Const(DateTime.Now))
        .Column(Column.Const(DateTime.Now))
        .Column(Column.Const(groupId))
        .Column("Id")
        .From("SysAdminUnit")
        .Where().Not().Exists(new Select(UserConnection)
            .Column("Id").
            From("SysUserInRole").As("ur").
            Where("ur", "SysUserId").IsEqual("SysAdminUnit", "Id").
            And("ur", "SysRoleId").IsEqual(Column.Parameter(groupId))
            as Select)
        .And("LDAPEntryId").In(query));
insertSelect.Execute();
```


CustomQuery

CustomQuery

```
string selectSqlText = string.Format(@"SELECT con.Id, con.Name
    FROM Contact con
    JOIN SysAdminUnit sau ON con.Id = sau.ContactId
    JOIN SysUserInRole suir ON sau.Id = suir.SysUserId
    WHERE suir.SysRoleId = '{0}'", approvalId );
var query = new CustomQuery(UserConnection, selectSqlText);
```

EntitySchemaQuery

EntitySchemaQuery

EntitySchemaQuery – это высокоуровневый класс, который предназначен для построения запросов на выборку из базы данных.

Особенности

Поддержка прав доступа

Механизм кеширования

Дополнительные настройки запроса

EntitySchemaQuery. Добавление колонок.

```
EntitySchema schema =
UserConnection.EntitySchemaManager.GetInstanceByName("Contact");
EntitySchemaQuery esq = new EntitySchemaQuery(schema);
// Добавить колонку первичного ключа
esq.PrimaryQueryColumn.IsAlwaysSelect = true;
// Колонка корневой схемы
string emailColumnName = esq.AddColumn("Email").Name;
// Колонка - справочник
string ownerColumnName = esq.AddColumn("Owner.Name").Name;
// Колонка схемы по произвольному внешнему ключу
string activityContactColumnName =
esq.AddColumn("[Activity:Owner:Id].Contact.Name").Name;
```

EntitySchemaQuery. Добавление колонок.

Колонка схемы по произвольному внешнему ключу

Имя колонки строится по принципу

[Имя_присоединяемой_схемы:Имя_колонки_для_связи_присоединяемой_схемы:Имя_колонки_для_связи_текущей_схемы].

Если в качестве колонки для связи у текущей схемы выступает колонка Id, то в этом случае ее можно опустить, то есть имя колонки будет выглядеть следующим образом:

[Имя_присоединяемой_схемы:Имя_колонки_для_связи_присоединяемой_схемы].

EntitySchemaQuery. Фильтрация.

Фильтр - это набор условий, применяемых при отображении данных запроса.

CreateFilter()

CreateFilterWithParameters()

CreateIsNullFilter()

CreateIsNotNullFilter()

CreateExistsFilter()

CreateNotExistsFilter()

```
esq.CreateFilterWithParameters(FilterComparisonType.Contain,  
"[SysUserInRole:SysUser].SysRole.Name", "Юрист"); // Юрист
```

EntitySchemaQuery. Группировка фильтров.

По умолчанию фильтры группируются через AND. Дополнительная группировка осуществляется через создание экземпляра класса [EntitySchemaQueryFilterCollection](#)

```
var filterOwner = esq.CreateFilterWithParameters(FilterComparisonType.Equal,
"Contact.Id", ownerId);
var filterLaywer = esq.CreateFilterWithParameters(FilterComparisonType.Contain,
"[SysUserInRole:SysUser].SysRole.Name", "Юрист"); // Юрист
EntitySchemaQueryFilterCollection filterCustom = new
EntitySchemaQueryFilterCollection(esq, LogicalOperationStrict.Or);
filterCustom.Add(filterOwner);
filterCustom.Add(filterLaywer);
esq.Filters.Add(esq.CreateFilterWithParameters(FilterComparisonType.Equal, "Id",
userConnection.CurrentUser.Id));
esq.Filters.Add(filterCustom);
```

EntitySchemaQuery. Агрегирующие функции.

Создание и добавление колонки в виде подзапроса , возвращающего результат указанной агрегирующей функции, по пути к колонке схемы относительно корневой RootSchema

```
var esq = new EntitySchemaQuery(EntitySchemaManager, "ESQCountryNA");  
EntitySchemaQuery subEsq = null;  
esq.AddColumn("[ESQCityNA:Country].Name", AggregationTypeStrict.Count, out subEsq);  
string sqlText = esq.GetSelectQuery(CurrentConnection).GetSqlText();
```

```
SELECT (  
    SELECT COUNT("SubESQCityNA"."Id") "Count"  
    FROM "ESQCityNA" "SubESQCityNA"  
    WHERE "SubESQCityNA"."CountryId" = "ESQCountryNA"."Id") "SubESQCityNA"  
FROM "ESQCountryNA" "ESQCountryNA"
```


EntitySchemaQuery. Использование макросов.

```
// Создание экземпляра запроса с корневой схемой Contact.
var esqContact = new EntitySchemaQuery(userConnection.EntitySchemaManager, "Contact");
// В качестве типа макроса выбирается тип [Текущий месяц].
var esqMacroType = EntitySchemaQueryMacroType.CurrentMonth;
// Создание экземпляра выражения для колонки "CreatedOn".
var esqRightExpression = new
EntitySchemaQueryExpression(esqContact.CreateSchemaColumnExpression("CreatedOn"));
// Создание экземпляра фильтра с заданными параметрами.
var esqFilter = esqContact.CreateFilter(FilterComparisonType.Equal, esqRightExpression,
esqMacroType);
// Добавление фильтра в коллекцию фильтров результирующего запроса.
esqContact.Filters.Add(esqFilter);
```



Россия:

443035 Самара,
Московское шоссе 4а

+7 (846) 266-55-69
info@terrasoft.ru

123090, Москва,
Ул. Каланчевская, д. 16

+7 (495) 280-16-80
info@terrasoft.ru

Украина:

03118, Киев, пр-т.
Краснозвездный, 115-А

+38 (044) 363-31-33
info@terrasoft.ua

UK:

PORTLAND HOUSE,
BRESSENDAN PLACE,
LONDON SW1E 5RS

+44 (2) 0338 40040
info@tscrm.com

Alexandria:

901 N Pitt Street, Suite 325
Alexandria

+1 (917) 383-27-70
info@bpmonline.com