

**Группа компаний "Информация и управление"**

***Инструментальные средства объектной  
разработки программных систем  
S\_Технология (SDK S\_Технология)***

**Включение в систему и использование  
таблиц, созданных пользователем**

**Редакция 2.0.0.X**

**Воронеж, 2018**

## АННОТАЦИЯ

Настоящая инструкция предназначена для администратора информационной системы, разработанной с использованием инструментальных средств объектной разработки программных систем *S\_Технология*, и содержит руководство по включению в систему новых таблиц, которые изначально не были предусмотрены разработчиками системы.

Приведены требования к оформлению новых таблиц, настройка системы для обработки данных в таких таблицах – редактирование данных в экранных формах, в отчетах, в операциях экспорта-импорта данных, в конструкторе запросов.

Инструментальные средства объектной разработки программных систем *S\_Технология* (SDK *S\_Технология*) разработаны группой компаний "Информация и управление" (компаниями "ИнфоМега" и "Информация и управление"), г. Воронеж.

Документ соответствует версии SDK *S\_Технология*, начиная с редакции 2.0.0.X, до выпуска новой редакции.

## СОДЕРЖАНИЕ

<b>ТЕРМИНЫ И СОКРАЩЕНИЯ.....</b>	<b>4</b>
<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>ПРОЕКТНЫЕ И ПОЛЬЗОВАТЕЛЬСКИЕ ТАБЛИЦЫ .....</b>	<b>4</b>
<b>ПОДКЛЮЧЕНИЕ К ПРОЕКТУ ТАБЛИЦ, СОЗДАННЫХ ПОЛЬЗОВАТЕЛЕМ .....</b>	<b>5</b>
<b>ИСПОЛЬЗОВАНИЕ ТАБЛИЦ, СОЗДАННЫХ ПОЛЬЗОВАТЕЛЕМ .....</b>	<b>9</b>
Экранные формы для таблиц, созданных пользователем .....	9
<i>Стандартные формы.....</i>	<i>10</i>
<i>Настраиваемые формы.....</i>	<i>12</i>
ИСПОЛЬЗОВАНИЕ ТАБЛИЦ, СОЗДАННЫХ ПОЛЬЗОВАТЕЛЕМ, В ОТЧЕТАХ.....	14
ИСПОЛЬЗОВАНИЕ ТАБЛИЦ, СОЗДАННЫХ ПОЛЬЗОВАТЕЛЕМ, В ОБМЕНЕ ДАННЫМИ .....	14
ИСПОЛЬЗОВАНИЕ ТАБЛИЦ, СОЗДАННЫХ ПОЛЬЗОВАТЕЛЕМ, В ЗАПРОСАХ К БД.....	15
<b>ПРИЛОЖЕНИЕ. ПРИМЕР СКРИПТОВ НА СОЗДАНИЕ ТАБЛИЦ, ПОСЛЕДОВАТЕЛЬНОСТЕЙ И ВЫДАЧУ ГРАНТОВ ...</b>	<b>16</b>
СУБД MS SQL .....	16
<i>R_REASON_OUTSOURCING .....</i>	<i>16</i>
<i>EXTERNAL_EXECUTOR_EXEX .....</i>	<i>17</i>
<i>EXTER_EXEC_LIST_JOB_POSI .....</i>	<i>19</i>
СУБД ORACLE.....	22
<i>R_REASON_OUTSOURCING .....</i>	<i>22</i>
<i>EXTERNAL_EXECUTOR_EXEX .....</i>	<i>23</i>
<i>EXTER_EXEC_LIST_JOB_POSI .....</i>	<i>25</i>

## Термины и сокращения

БД	База данных
СУБД	Система управления базами данных

## Введение

Инструментальные средства объектной разработки программных систем *S\_Технология* предназначены для разработки информационных систем любой сложности на платформе .Net в среде разработки Microsoft Visual Studio на языке C# для архитектур Windows Forms и Web.

## Проектные и пользовательские таблицы

Все таблицы любого проекта *S\_Технологии* по происхождению делятся на **проектные таблицы** (предусмотренные разработчиком на этапе проектирования системы или разработчиком *S\_Технологии*) и **таблицы, созданные пользователем** (добавленные в систему конечными пользователями без участия разработчика).

Инструментальные средства разработки *S\_Технология* дают возможность конечным пользователям добавлять в проект новые таблицы. Смысл добавления в том, что сами пользователи, без участия разработчиков, могут расширять состав проекта. При этом обработка новых пользовательских таблиц в системе принципиально не отличается от обработки проектных таблиц. По сути, единственное ограничение в том, что проектные таблицы не могут содержать ссылки на новые таблицы, так как во время проектирования ничего не знают о них. Но сама обработка пользовательских таблиц выполняется практически так же, как и для проектных таблиц: данные в пользовательских таблицах могут редактироваться в таких же экранных формах (таблицы "видит" Генератор форм), можно создавать отчеты Генератором отчетов, выполнять экспорт-импорт данных пользовательских таблиц модулем обмена данных, конструировать запросы в Конструкторе запросов.

Общий порядок обработки пользовательских таблиц следующий.

- Пользователь создает новые таблицы (общепринятыми средствами), причем есть возможность указать русские имена таблиц и полей таблиц.
- Запускается генерация модуля таблицы на языке C#. Сгенерированный модуль (содержит класс сущности со стандартными методами и дата-класс, также со стандартными методами) подключается к системе.
- Во время запуска системы модуль транслируется "на лету", и протранслированный модуль становится виден всем компонентам системы.
- Стандартные экранные формы генерируются также "на лету" Генератором форм, и есть возможность в Генераторе форм настроить и получить сложные экранные формы.
- Класс сущности доступен для обработки в других модулях SDK *S\_Технология* – в Генераторе отчетов, в модуле обмена данных, в Конструкторе запросов.

**Фактически SDK *S\_Технология* предоставляет все основные компоненты для автоматизированной разработки модулей без участия разработчиков.**

## Подключение к проекту таблиц, созданных пользователем

Чтобы добавить новые таблицы в систему без участия разработчика, необходимо выполнить следующие действия.

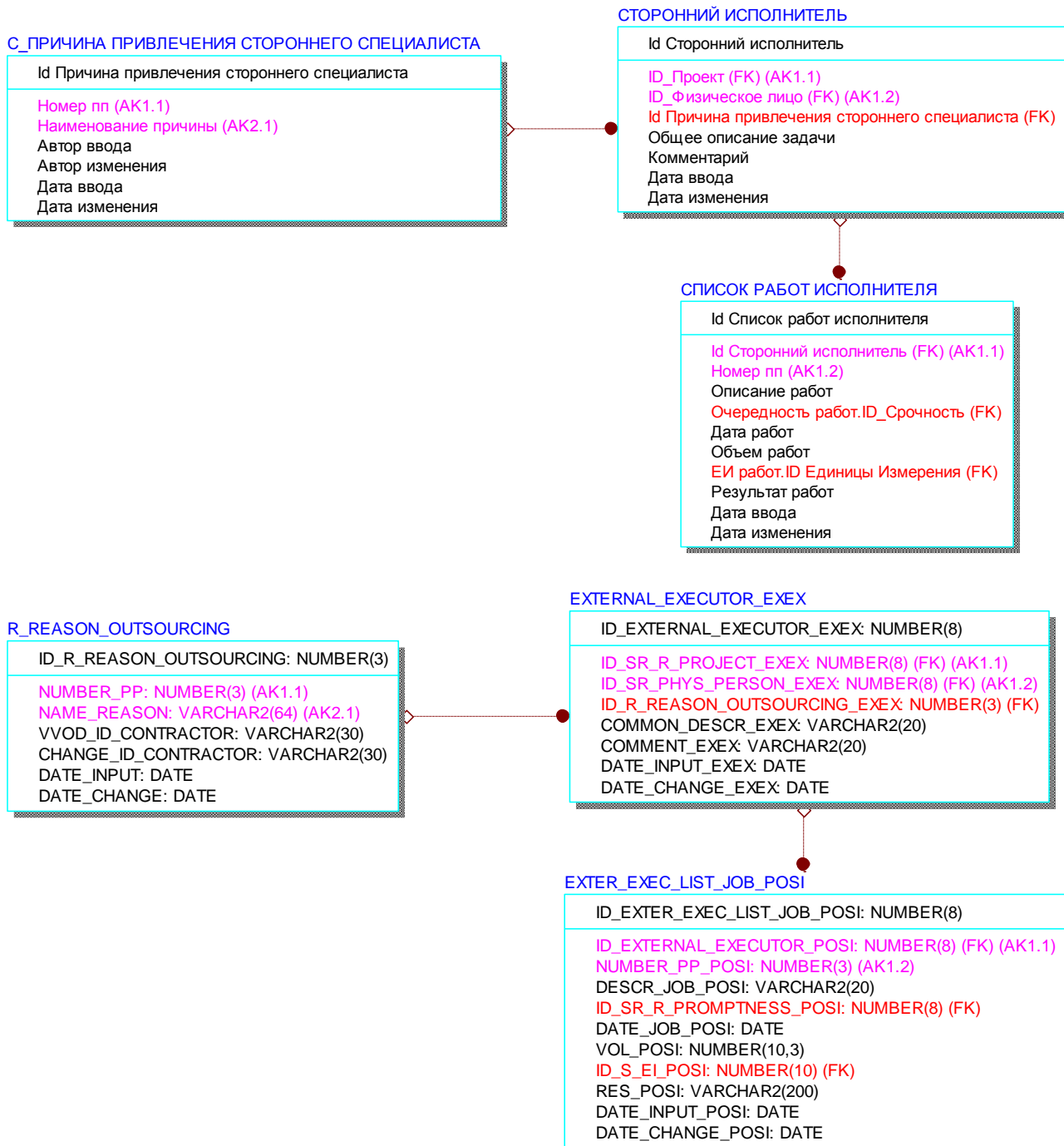
1. Средствами базы данных создать нужные таблицы в БД проекта. Для СУБД Oracle это можно сделать, например, с помощью программ «sqldeveloper» или «TOAD»; для СУБД MS SQL – с помощью программы «SQL Server Management Studio». Таблицы должны отвечать перечисленным ниже требованиям.
  - Таблицы, созданные пользователем, должны иметь первичный ключ (ID), состоящий из одного целочисленного поля. Физическое имя первичного ключа должно иметь формат: **ID\_<физическое имя таблицы>**. Например, если таблица называется *NEW\_TABLE*, то первичный ключ должен называться *ID\_NEW\_TABLE*.
  - Таблицы, созданные пользователем, могут иметь один или несколько альтернативных ключей (ключей уникальности), состоящих из любого числа полей любого формата.
  - Таблицы, созданные пользователем, могут иметь один или несколько внешних ключей (ссылок). Ссылки могут быть как на другие таблицы, созданные пользователем, так и на любые проектные таблицы.
  - Если необходимо, чтобы таблицы и их поля имели русские наименования в проекте, то при создании таблиц в БД русские наименования таблиц и полей следует сохранить в комментариях к таблицам и полям соответственно.
  - Для таблиц, созданных пользователем, должны быть выданы права (гранты) ALTER, DELETE, INDEX, INSERT, SELECT, UPDATE, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, DEBUG, FLASHBACK для пользователей CUSTOM, RTIME\_DBA, S\_DBA.
  - Таблицы, созданные пользователем, могут иметь учетные поля. Если эти поля есть в таблице и их физические имена соответствуют шаблону (см. таблицу ниже), то эти поля будут заполняться и обрабатываться автоматически, по общим правилам, действующим для всех проектных таблиц.

Назначение поля	Шаблон физического имени поля	Комментарий
Автор ввода	AUTHOR_INPUT_<суффикс>	Суффикс – необязательная часть. Если таблица, созданная пользователем, суффикса не имеет, то его не должно быть и в имени поля.
Автор изменения	AUTHOR_CHANGE_<суффикс>	Суффикс – необязательная часть. Если таблица, созданная пользователем, суффикса не имеет, то его не должно быть и в имени поля.
Дата ввода	DATE_INPUT_<суффикс>	Суффикс – необязательная часть. Если таблица, созданная пользователем, суффикса не имеет, то его не должно быть и в имени поля.
Дата изменения	DATE_CHANGE_<суффикс>	Суффикс – необязательная часть. Если таблица, созданная пользователем, суффикса не имеет, то его не должно быть и в имени поля.

Для примера, предположим, что пользователь создал три таблицы, связанные между собой:

- С\_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА (R\_REASON\_OUTSOURCING)
- СТОРОННИЙ ИСПОЛНИТЕЛЬ (EXTERNAL\_EXECUTOR\_EXEX)
- СПИСОК РАБОТ ИСПОЛНИТЕЛЯ (EXTER\_EXEC\_LIST\_JOB\_POSI)

Ниже на рисунках приведены диаграммы, показывающие поля этих таблиц и их связи между собой.



Следует обратить внимание, что помимо ссылок таблиц, созданных пользователем, друг на друга, таблица СТОРОННИЙ ИСПОЛНИТЕЛЬ имеет ссылки на проектные таблицы СЗ\_С\_ПРОЕКТ (SR\_R\_PROJECT\_SRPJ) и СЗ\_ФИЗИЧЕСКОЕ ЛИЦО (SR\_PHYS\_PERSON\_SRPP), а таблица СПИСОК РАБОТ ИСПОЛНИТЕЛЯ имеет ссылку на проектные справочники СЗ\_С\_СРОЧНОСТЬ (SR\_R\_PROMPTNESS\_SRPR) и С\_ЕДИНИЦА ИЗМЕРЕНИЯ (S\_EI\_EIEI).

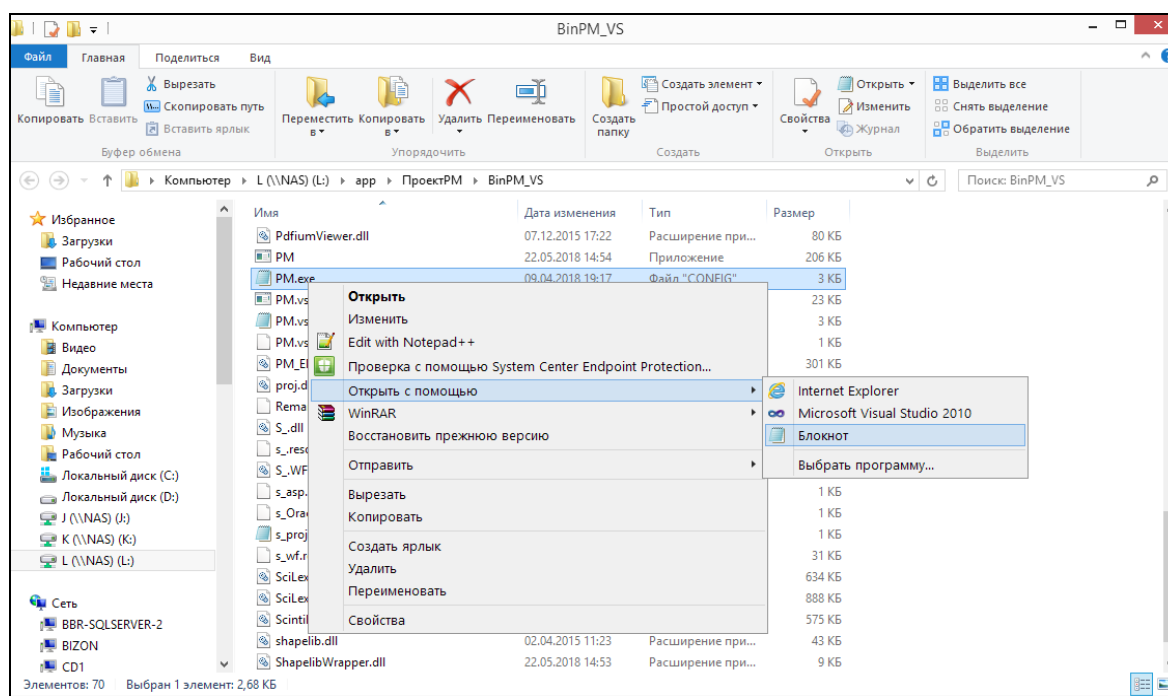
В приложении приведены скрипты на создание таблиц С\_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА, СТОРОННИЙ ИСПОЛНИТЕЛЬ и СПИСОК РАБОТ ИСПОЛНИТЕЛЯ, и выдачу грантов на эти таблицы в базах MS SQL и ORACLE.

2. Средствами базы данных создать последовательности для новых таблиц, соблюдая правила и ограничения, приведенные ниже.

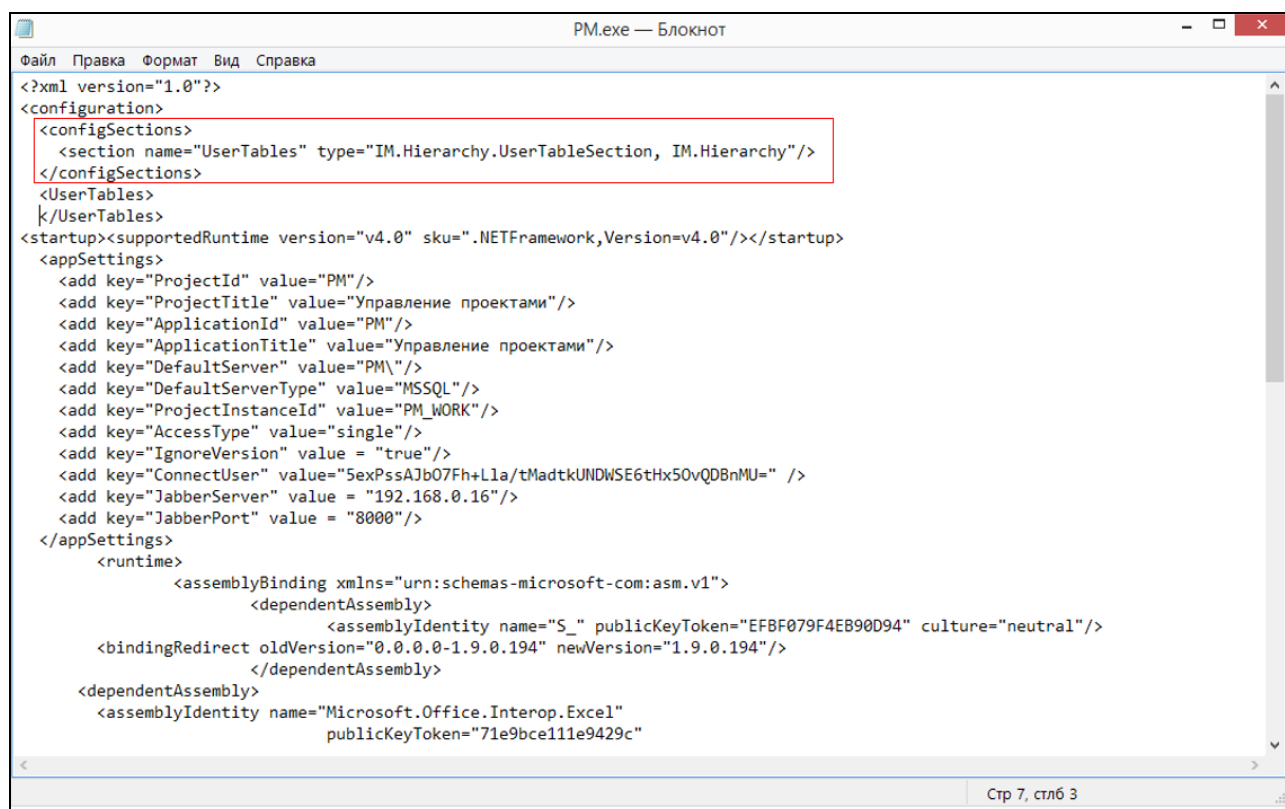
- Последовательность должна создаваться в той же схеме, что и таблица.
- Название последовательности должно иметь формат **SEQ\_<физическое имя таблицы>**. Например, если таблица называется *NEW\_TABLE*, то последовательность должна называться *SEQ\_NEW\_TABLE*.
- У последовательности должны быть права (гранты) на чтение для пользователя PUBLIC.
- Ограничений на шаг, начальное и максимальное значение последовательности нет.

В приложении приведены скрипты на создание последовательностей таблиц С\_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА, СТОРОННИЙ ИСПОЛНИТЕЛЬ и СПИСОК РАБОТ ИСПОЛНИТЕЛЯ в базах MS SQL и ORACLE.

3. Открыть файл конфигурации проекта <Идентификатор проекта>.exe.config (он лежит в папке программ проекта). Например, если идентификатор проекта «РМ», то файл будет иметь имя *PM.exe.config*. Открыть этот файл можно с помощью блокнота или иной подходящей программы.



Проверить в файле конфигурации наличие секции UserTables внутри блока <configSections> (этот блок должен быть первым внутри блока <configuration>).



```

PM.exe — Блокнот
Файл Правка Формат Вид Справка
<?xml version="1.0"?>
<configuration>
  <configSections>
    <section name="UserTables" type="IM.Hierarchy.UserTableSection, IM.Hierarchy"/>
  </configSections>
  <UserTables>
  </UserTables>
  <startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/></startup>
  <appSettings>
    <add key="ProjectId" value="PM"/>
    <add key="ProjectTitle" value="Управление проектами"/>
    <add key="ApplicationId" value="PM"/>
    <add key="ApplicationTitle" value="Управление проектами"/>
    <add key="DefaultServer" value="PM\"/>
    <add key="DefaultServerType" value="MSSQL"/>
    <add key="ProjectInstanceId" value="PM_WORK"/>
    <add key="AccessType" value="single"/>
    <add key="IgnoreVersion" value = "true"/>
    <add key="ConnectUser" value="5exPssAjb07Fh+Lla/tMadtkUNDWSE6tHx50vQDBnMU=" />
    <add key="JabberServer" value = "192.168.0.16"/>
    <add key="JabberPort" value = "8000"/>
  </appSettings>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="S_" publicKeyToken="EFBF079F4EB90D94" culture="neutral"/>
        <bindingRedirect oldVersion="0.0.0.0-1.9.0.194" newVersion="1.9.0.194"/>
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="Microsoft.Office.Interop.Excel"
          publicKeyToken="71e9bce111e9429c"
        >

```

Если блока нет, то добавить его:

```

<configSections>
  <section name="UserTables" type="IM.Hierarchy.UserTableSection, IM.Hierarchy"/>
</configSections>

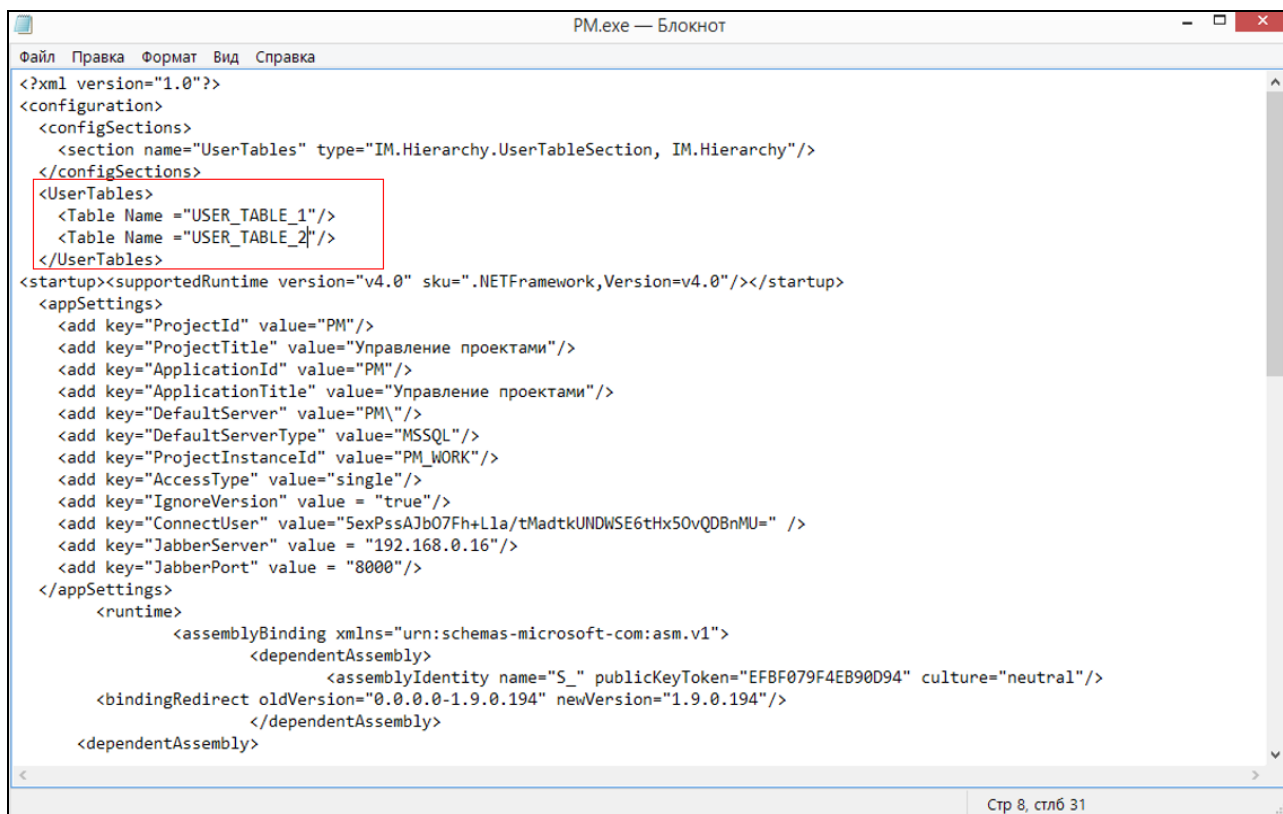
```

Найти в файле конфигурации блок <UserTables> (если блока нет, то добавить) и перечислить в этом блоке физические имена новых таблиц, созданных пользователем, в формате:

<Table Name = "*физическое имя таблицы*" />

Например:





Т.е. чтобы подключить к проекту три таблицы, описанные выше, нужно добавить следующий текст в блок <UserTables>:

```

<UserTables>
  <Table Name ="R_REASON_OUTSOURCING"/>
  <Table Name ="EXTERNAL_EXECUTOR_EXEX"/>
  <Table Name ="EXTER_EXEC_LIST_JOB_POSI"/>
</UserTables>

```

4. Запустить проект. При старте программа автоматически сформирует сущности для всех таблиц, созданных пользователем в базе и перечисленных в блоке UserTable файла конфигурации.

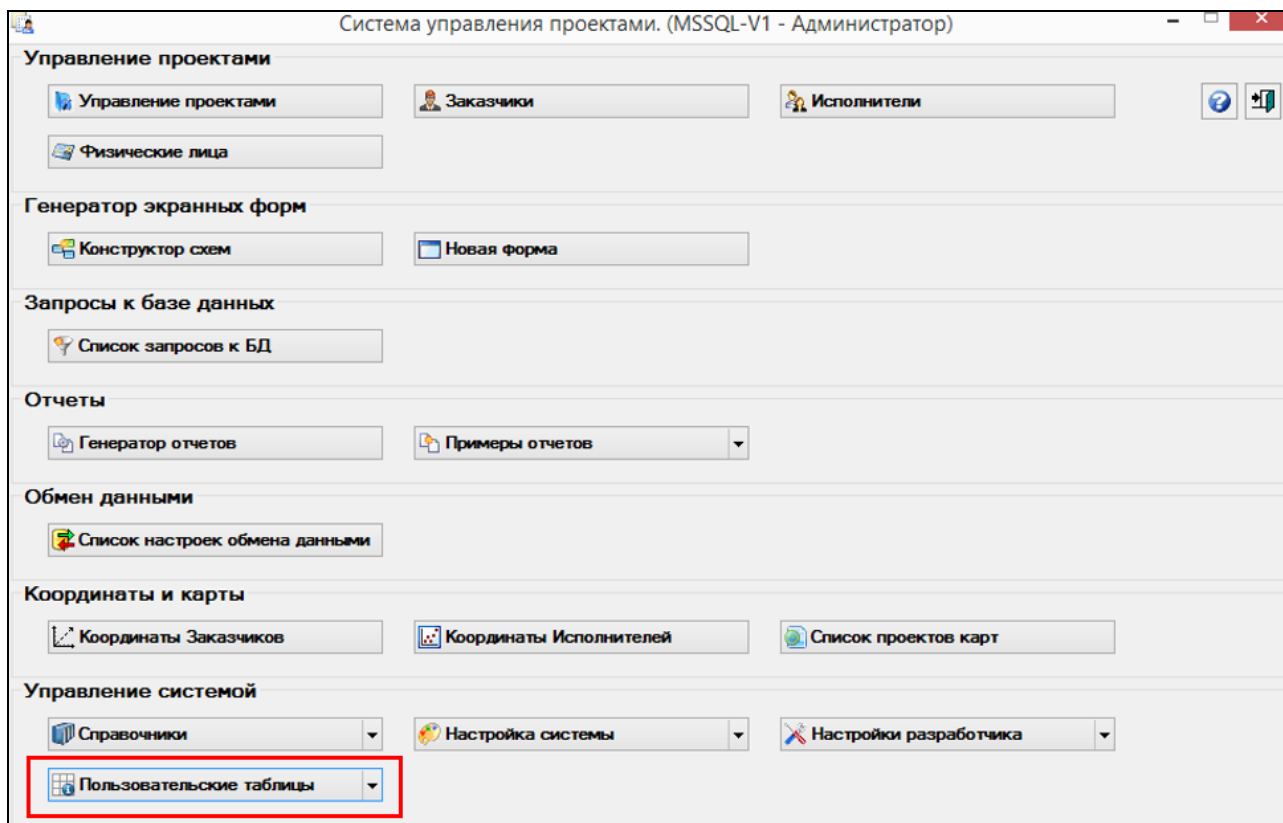
Если таблица создана в базе, но ее описание отсутствует в файле конфигурации, то она не будет подключена к проекту.

Если таблица не создана в базе, но описана в файле конфигурации, при старте программы будет сообщение об ошибке: [s\_2238E] Объект БД с именем "USER\_TABLE" не найден.

## Использование таблиц, созданных пользователем

### Экранные формы для таблиц, созданных пользователем

Все таблицы, созданные пользователем, могут быть открыты для просмотра и редактирования в меню *Управление системой – Пользовательские таблицы*. Пункт меню появится после соответствующей настройки, см. разделы ниже. Кроме того, пункт меню закрыт привилегией *CALL\_MENU\_USER\_FORM* Пункт меню *Пользовательские таблицы*.



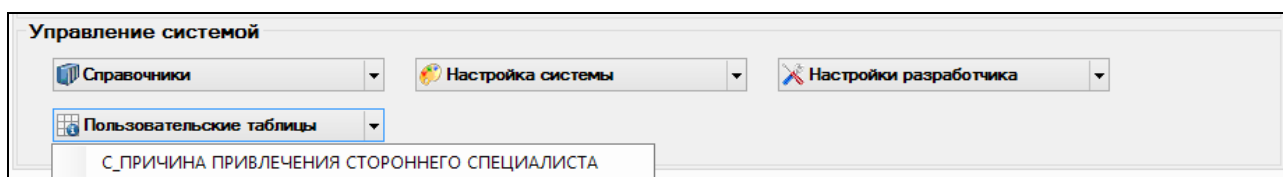
Этот пункт содержит все формы подключенных к проекту таблиц пользователей, как [стандартные](#) (создаются системой автоматически), так и [настроенные](#) администраторами системы.

### Стандартные формы

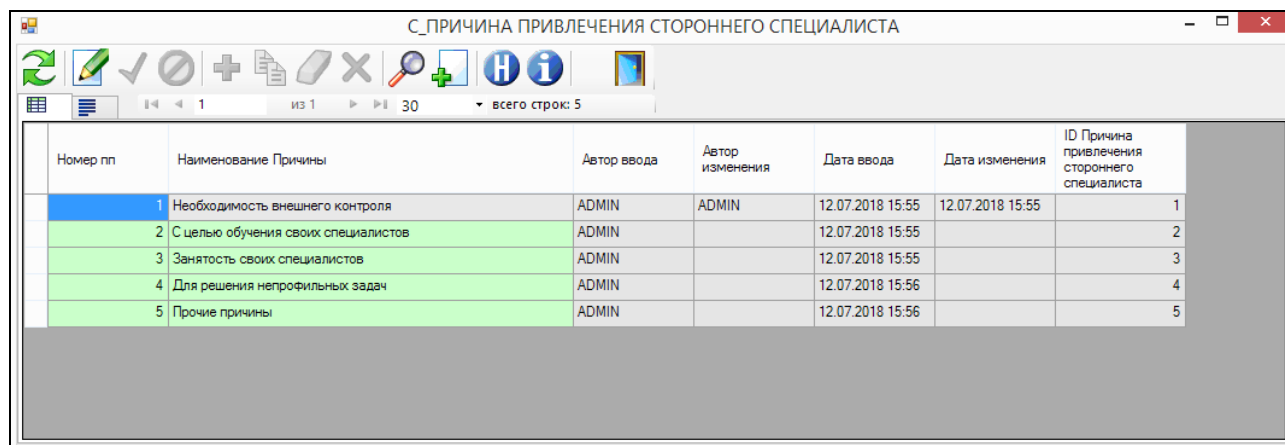
Для того, чтобы для таблицы, созданной пользователем, система автоматически сформировала стандартную форму и появился пункт меню для ее вызова, следует отметить эту таблицу в конфигурации специальным тегом *AutoForm* со значением «1». Например, чтобы в стандартной форме открывался справочник *С\_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА* необходимо в конфигурации в блоке *UserTables* в строке таблицы этого справочника добавить текст *AutoForm* = «1».

```
<UserTables>
  <Table Name ="R_REASON_OUTSOURCING" AutoForm = "1"/>
  <Table Name ="EXTERNAL_EXECUTOR_EXEX"/>
  <Table Name ="EXTER_EXEC_LIST_JOB_POSI"/>
</UserTables>
```

После перезапуска системы, можно увидеть, что в меню *Управление системой* – *Пользовательские таблицы* появился пункт для вызова справочника причин привлечения стороннего специалиста в стандартной форме.



Ниже приведен вид пользовательской таблицы в автоматической стандартной форме списка.



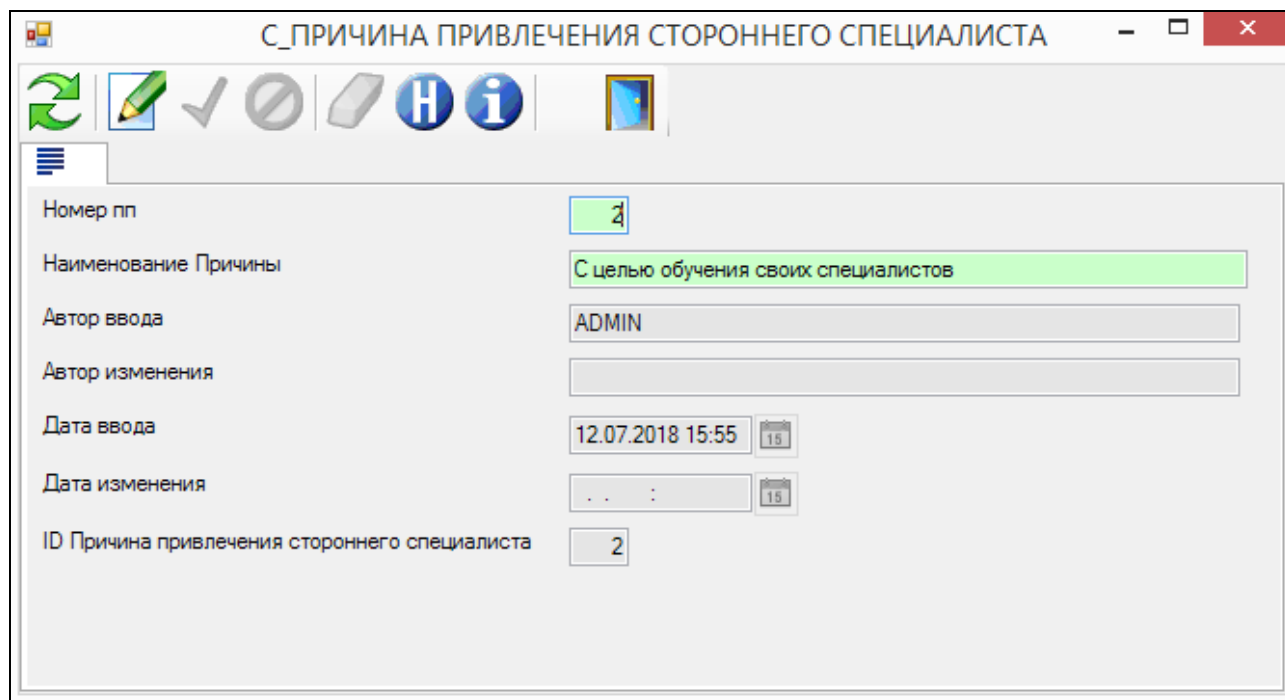
Номер пп	Наименование Причины	Автор ввода	Автор изменения	Дата ввода	Дата изменения	ID Причина привлечения стороннего специалиста
1	Необходимость внешнего контроля	ADMIN	ADMIN	12.07.2018 15:55	12.07.2018 15:55	1
2	С целью обучения своих специалистов	ADMIN		12.07.2018 15:55		2
3	Занятость своих специалистов	ADMIN		12.07.2018 15:55		3
4	Для решения непрофильных задач	ADMIN		12.07.2018 15:56		4
5	Прочие причины	ADMIN		12.07.2018 15:56		5

Форма состоит из панели управления и одного списка.

На панели управления представлены все стандартные кнопки формы списка проектов на базе S\_Технологии: Обновить, Начать редактирование, Сохранить изменения, Отменить изменения, Добавить новую запись, Копировать запись, Очистить ячейку записи, Удалить запись, Вызвать карточку записи, Добавить новую запись через карточку и другие.

В списке доступны все стандартные сервисы сеток проектов на базе S\_Технологии: перестановка колонок, сортировка по нажатию на заголовок колонки, контекстный фильтр по колонке, функции меню правой кнопки мыши (сложные фильтры и сортировки, управление видимостью колонок, экспорт в excel и др.).

Ниже приведен вид одной строки пользовательской таблицы в автоматической стандартной форме карточки.



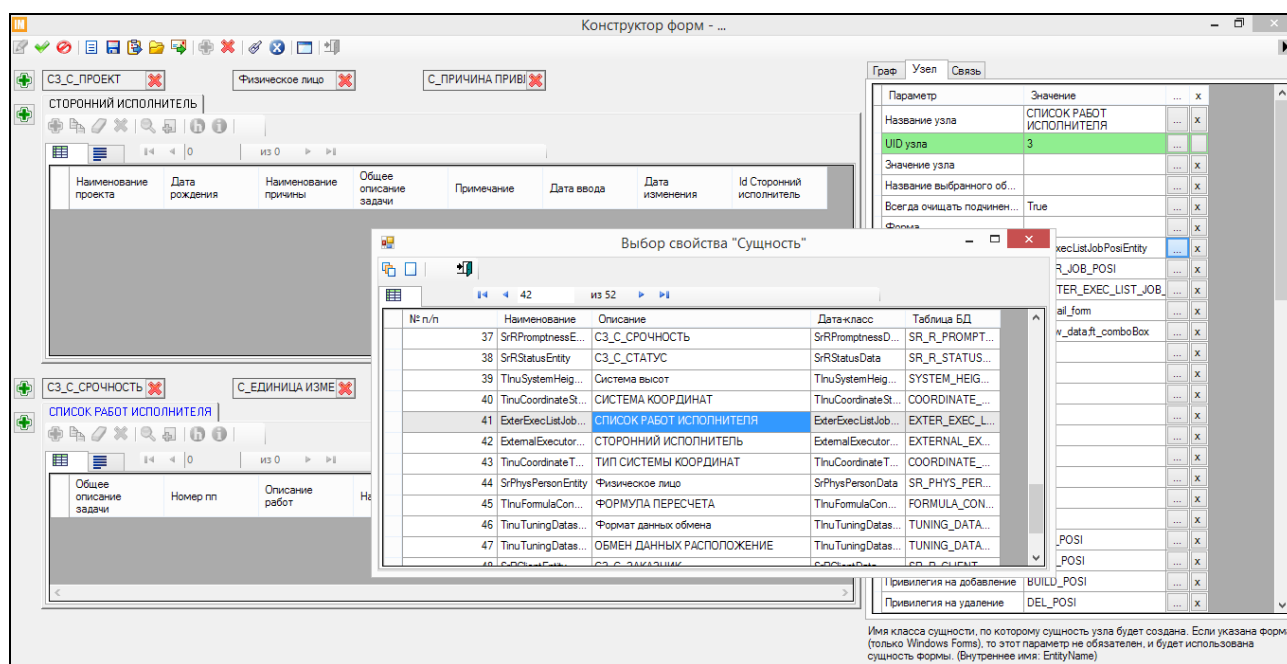
Номер пп	2
Наименование Причины	С целью обучения своих специалистов
Автор ввода	ADMIN
Автор изменения	
Дата ввода	12.07.2018 15:55
Дата изменения	
ID Причина привлечения стороннего специалиста	2

К автоматическим формам для таблиц, созданным пользователем, применяются все настройки текущего пользователя: сохранение конфигурации, цвет обязательных и read-only колонок, опции «Основной интерфейс» (отключить редактирование в списке) и «Увеличенный размер кнопок».

Стандартные (автоматические) формы отображают данные только одной таблицы, созданной пользователем. Если необходимо отображать данные нескольких таблиц, созданных пользователем, в общей форме (мастер-деталь, условия отбора), то необходимо средствами генератора экранных форм настроить и сохранить форму в папке *UserSchemes* (внутри папки программ проекта).

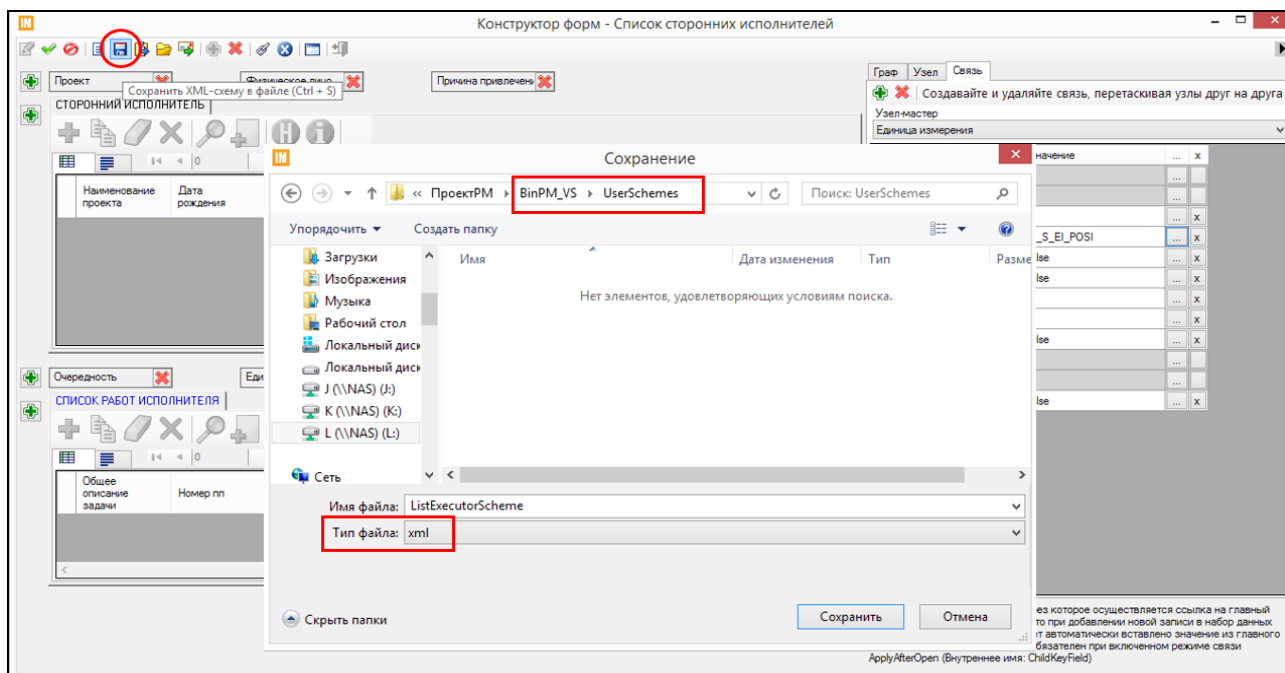
## Настраиваемые формы

С помощью инструмента *S\_Технологии* «Генератор экранных форм» пользователь имеет возможность создавать более сложные формы для таблиц, созданных пользователем (мастер-деталь, условия отбора и т.д.). Расположение пункта для вызова генератора форм в меню конкретного проекта определяется разработчиком этого проекта, пункт доступен по привилегии *Call\_Graf\_Constructor* Пункт меню "Конструктор схем".

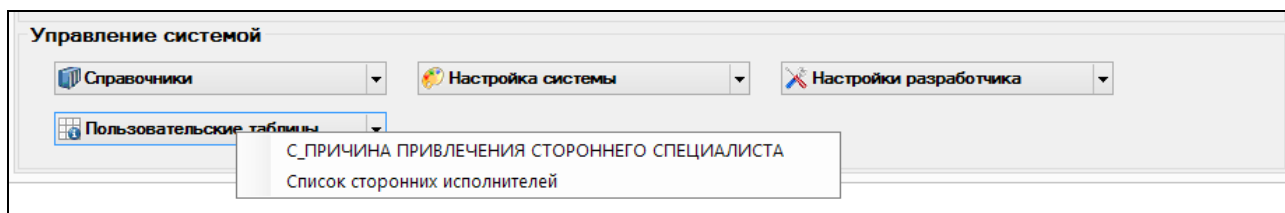


Настройка форм детально описана в отдельном томе «ИМ Порядок создания приложения Windows Forms. Состав проекта».

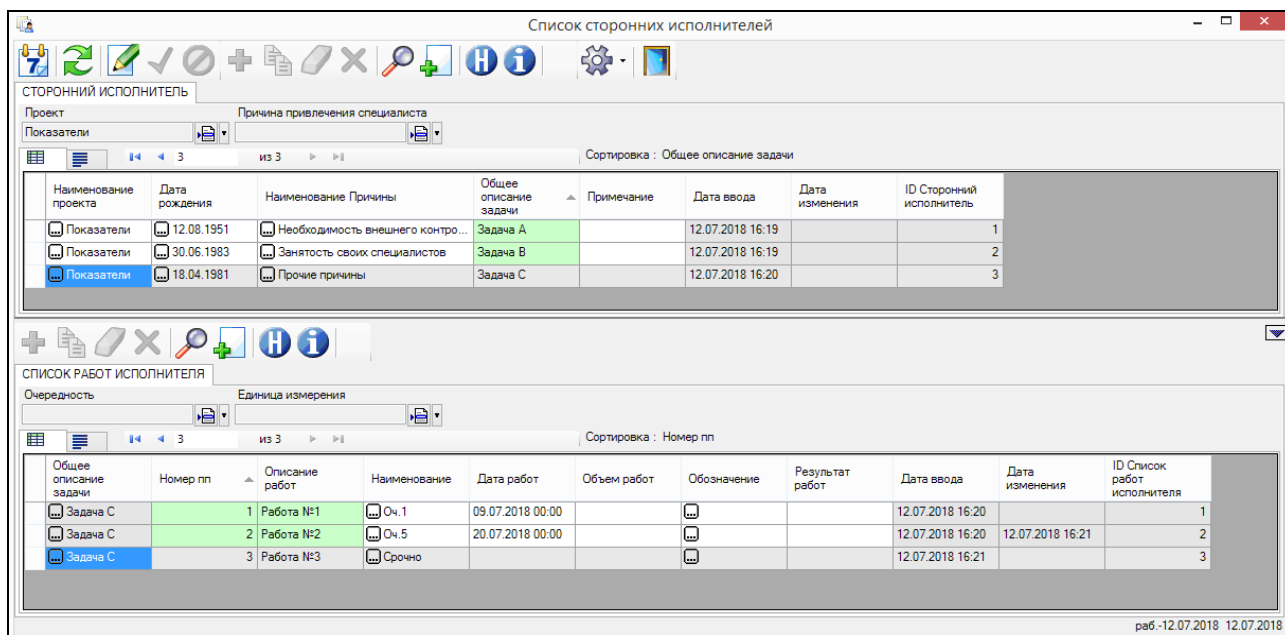
Готовые формы нужно сохранять в формате xml, в папке бина *UserSchemes*.



Формы, сохраненные пользователем в папке *UserSchemes*, будут доступны в меню *Управление системой – Пользовательские таблицы* наряду с автоматическими формами.

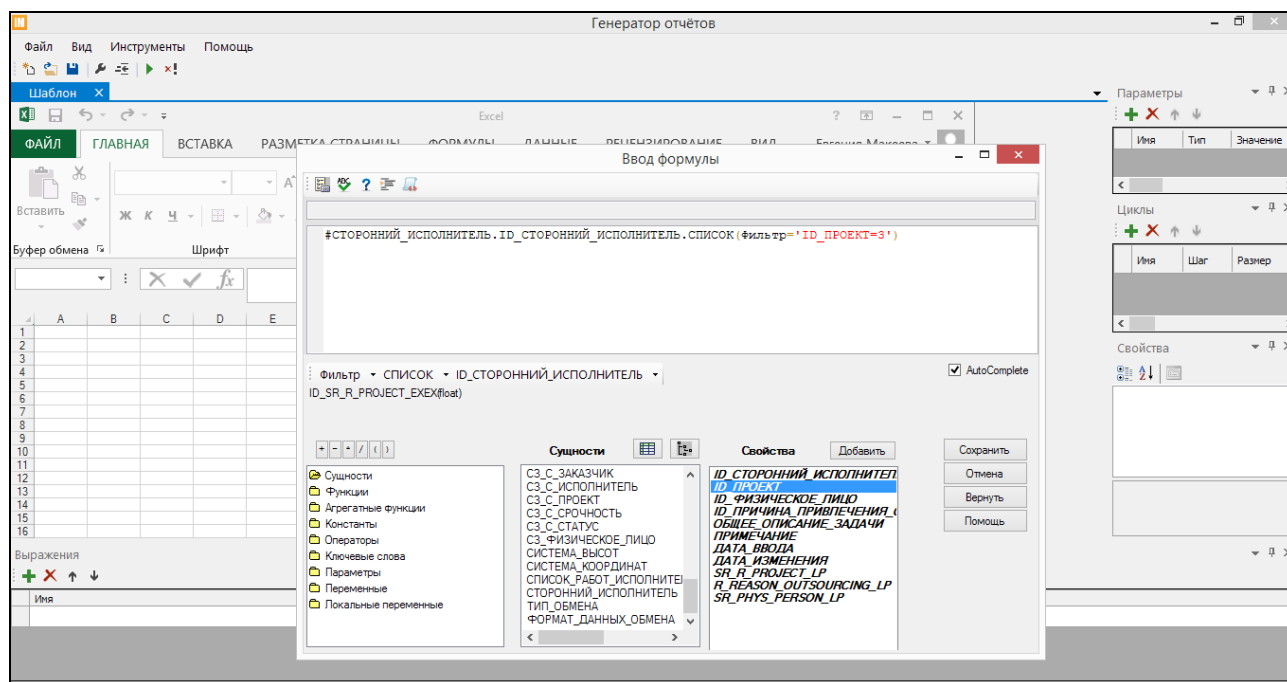


Ниже приведен пример настроенной формы, отображающей данные двух таблиц, созданных пользователем с условиями отборов. Сетка мастера формы отображает список сторонних исполнителей (таблица **СТОРОННИЙ ИСПОЛНИТЕЛЬ**), а детальная сетка отображает работы каждого исполнителя (таблица **СПИСОК РАБОТ ИСПОЛНИТЕЛЯ**).



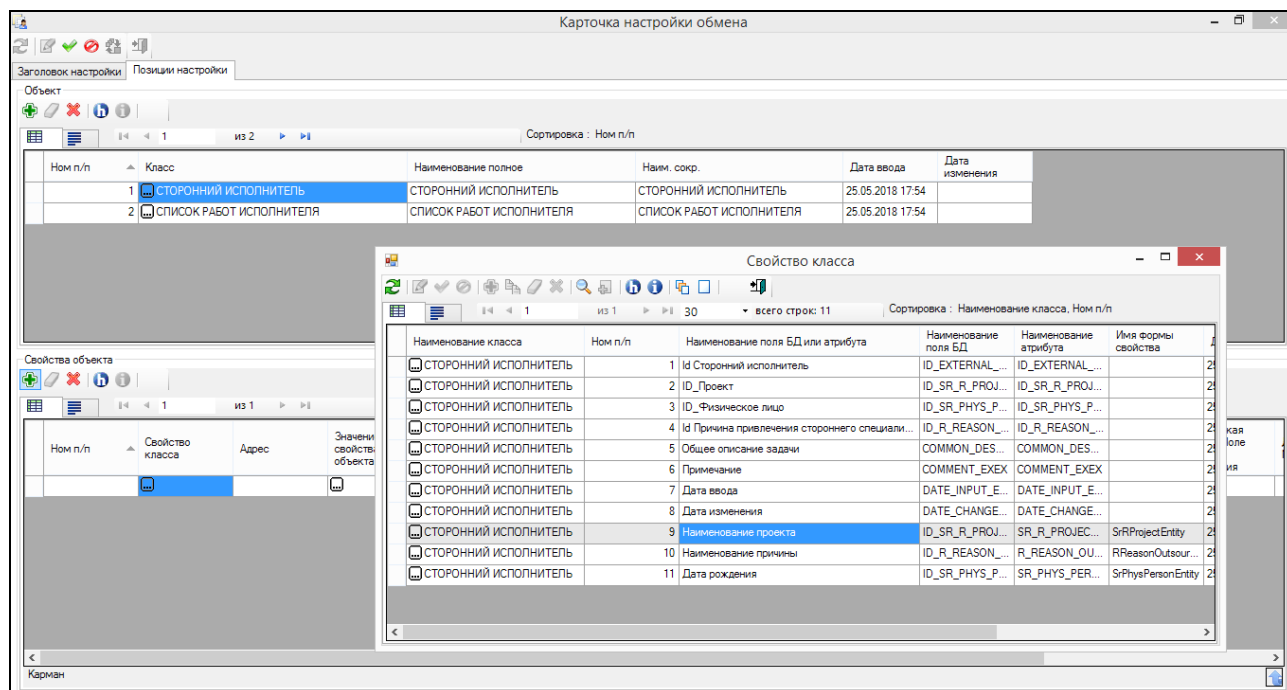
## Использование таблиц, созданных пользователем, в отчетах

Пользователь может использовать таблицы, созданные пользователем, для создания отчетов (шаблоны генератора отчетов) так же, как проектные таблицы.



## Использование таблиц, созданных пользователем, в обмене данными

Пользователь может использовать таблицы, созданные пользователем, в настройках обмена данными так же, как проектные таблицы. Можно выполнять как импорт данных в таблицы, созданные пользователем, так и экспорт данных из них.



В текущей версии инструментов S\_Технологии существует ограничение на использование таблиц, созданных пользователем, в обмене данными. Для того чтобы таблицу можно было включить в настройку обмена, требуется чтобы описание этой таблицы и ее полей присутствовало в системных справочниках ОПИСАНИЕ КЛАССА и СВОЙСТВО КЛАССА. Право на пополнение этих справочников пока есть только у разработчика системы, поэтому самостоятельно пользователь не сможет загрузить описания своих таблиц в справочники. В следующих версиях такую возможность планируется добавить.

Примечание. Если в проекте обмен данными выполняется не в основном приложении, а в отдельном приложении обмена данными, то для использования таблиц, созданных пользователем, в обмене данными, необходимо добавить физические наименования этих таблиц в блок UserTables файла конфигурации приложения обмена данными (так же, как это было сделано в файле конфигурации основного приложения, см. главу [Подключение к проекту таблиц, созданных пользователем](#)).

### Использование таблиц, созданных пользователем, в запросах к БД

Пользователь может использовать таблицы, созданные пользователем, для формирования запросов к базе данных, так же, как проектные таблицы.

№ n/n	Наименование	Описание
24	СПИСОК РАБОТ ИСПОЛНИТЕЛЯ	ЕИ работ
25	СПИСОК РАБОТ ИСПОЛНИТЕЛЯ	Результат работ
26	СПИСОК РАБОТ ИСПОЛНИТЕЛЯ	Дата ввода
27	СПИСОК РАБОТ ИСПОЛНИТЕЛЯ	Дата изменения
28	С_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА	Id Причина привлечения стороннего специалиста
29	С_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА	Номер пп
30	С_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА	Наименование причины
31	С_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА	Автор ввода
32	С_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА	Автор изменения



## Приложение. Пример скриптов на создание таблиц, последовательностей и выдачу грантов

### СУБД MS SQL

Перед запуском скриптов нужно указать имя БД вместо текста <Имя БД>.

#### R\_REASON\_OUTSOURCING

```
USE [<ИМЯ БД>]
GO
```

```
/****** Object: Table [dbo].[R_REASON_OUTSOURCING]    Script Date: 23.05.2018 16:20:22
*****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[R_REASON_OUTSOURCING](
    [ID_R_REASON_OUTSOURCING] [numeric](3, 0) NOT NULL,
    [NUMBER_PP] [numeric](3, 0) NOT NULL,
    [NAME_REASON] [varchar](64) NOT NULL,
    [VVOID_ID_CONTRACTOR] [varchar](30) NOT NULL,
    [CHANGE_ID_CONTRACTOR] [varchar](30) NULL,
    [DATE_INPUT] [datetime] NOT NULL,
    [DATE_CHANGE] [datetime] NULL,
    CONSTRAINT [XPKR_REASON_OUTSOURCING] PRIMARY KEY NONCLUSTERED
(
    [ID_R_REASON_OUTSOURCING] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK1R_REASON_OUTSOURCING] UNIQUE NONCLUSTERED
(
    [NUMBER_PP] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK2R_REASON_OUTSOURCING] UNIQUE NONCLUSTERED
(
    [NAME_REASON] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
GO
```

```
ALTER TABLE [dbo].[R_REASON_OUTSOURCING] ADD DEFAULT (getdate()) FOR [DATE_INPUT]
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SHEMA', @level0name=N'dbo', @level1type=N'TABLE',
@level1name=N'R_REASON_OUTSOURCING',
@value=N'С_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА'
```



```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'R_REASON_OUTSOURCING',
@level2name=N'ID_R_REASON_OUTSOURCING',
@value=N'ID Причина привлечения стороннего специалиста'
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'R_REASON_OUTSOURCING',
@level2name=N'NUMBER_PP',
@value=N'Номер пп'
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'R_REASON_OUTSOURCING',
@level2name=N'NAME_REASON',
@value=N'Наименование Причины'
```

```
declare @id bigint;
use <ИМЯ БД>;
if exists (select 'x' from sysobjects obj where obj.name = 'R_REASON_OUTSOURCING' and
obj.type = 'U')
    select @id = ISNULL(MAX(t.ID_R_REASON_OUTSOURCING), 0) from R_REASON_OUTSOURCING t;
else
    set @id = 1;
use PM_INSTALL_S_DATA;
if exists(select 'x' from S_TB_SEQ seq where seq.SEQ_NAME = 'SEQ_R_REASON_OUTSOURCING')
    update S_TB_SEQ set SEQ_VALUE = @id, INCREMENT_BY = 1
    where SEQ_NAME = 'SEQ_R_REASON_OUTSOURCING';
else
    insert S_TB_SEQ (SEQ_NAME, INCREMENT_BY, SEQ_VALUE)
    values ('SEQ_R_REASON_OUTSOURCING', 1, @id);

go
```

## EXTERNAL\_EXECUTOR\_EXEX

```
USE [<ИМЯ БД>]
GO
```

```
/****** Object: Table [dbo].[EXTERNAL_EXECUTOR_EXEX]    Script Date: 23.05.2018 16:19:58
******/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX](
    [ID_EXTERNAL_EXECUTOR_EXEX] [numeric](8, 0) NOT NULL,
    [ID_SR_R_PROJECT_EXEX] [int] NULL,
    [ID_SR_PHYS_PERSON_EXEX] [int] NULL,
    [ID_R_REASON_OUTSOURCING_EXEX] [numeric](3, 0) NULL,
    [COMMON_DESCR_EXEX] [varchar](20) NOT NULL,
    [COMMENT_EXEX] [varchar](20) NULL,
    [DATE_INPUT_EXEX] [datetime] NOT NULL,
    [DATE_CHANGE_EXEX] [datetime] NULL,
    CONSTRAINT [XPKEEXTERNAL_EXECUTOR_EXEX] PRIMARY KEY NONCLUSTERED
(
    [ID_EXTERNAL_EXECUTOR_EXEX] ASC
```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK1EXTERNAL_EXECUTOR_EXEX] UNIQUE NONCLUSTERED
(
    [ID_SR_R_PROJECT_EXEX] ASC,
    [ID_SR_PHYS_PERSON_EXEX] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] ADD DEFAULT (getdate()) FOR [DATE_INPUT_EXEX]
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] WITH CHECK ADD CONSTRAINT [R_2704] FOREIGN
KEY([ID_R_REASON_OUTSOURCING_EXEX])
REFERENCES [dbo].[R_REASON_OUTSOURCING] ([ID_R_REASON_OUTSOURCING])
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] CHECK CONSTRAINT [R_2704]
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] WITH CHECK ADD CONSTRAINT [R_2853] FOREIGN
KEY([ID_SR_R_PROJECT_EXEX])
REFERENCES [dbo].[SR_R_PROJECT_SRPJ] ([ID_SR_R_PROJECT_SRPJ])
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] CHECK CONSTRAINT [R_2853]
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] WITH CHECK ADD CONSTRAINT [R_2854] FOREIGN
KEY([ID_SR_PHYS_PERSON_EXEX])
REFERENCES [dbo].[SR_PHYS_PERSON_SRPP] ([ID_SR_PHYS_PERSON_SRPP])
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] CHECK CONSTRAINT [R_2854]
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] WITH CHECK ADD CONSTRAINT
[EXTERNAL_EXECUTOR_EXEX__COMMENT_EXEX] CHECK (([COMMENT_EXEX]='H' OR [COMMENT_EXEX]='Д'))
GO

ALTER TABLE [dbo].[EXTERNAL_EXECUTOR_EXEX] CHECK CONSTRAINT
[EXTERNAL_EXECUTOR_EXEX__COMMENT_EXEX]
GO

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SHEMA', @level0name=N'dbo', @level1type=N'TABLE',
@level1name=N'EXTERNAL_EXECUTOR_EXEX',
@value=N'СТОРОННИЙ ИСПОЛНИТЕЛЬ'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTERNAL_EXECUTOR_EXEX',
@level2name=N'ID_EXTERNAL_EXECUTOR_EXEX',
@value=N'ID Сторонний исполнитель'

EXEC sys.sp_addextendedproperty @name=N'Comment',

```

```

@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTERNAL_EXECUTOR_EXEX',
@level2name=N'ID_SR_R_PROJECT_EXEX',
@value=N'ID Проект'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTERNAL_EXECUTOR_EXEX',
@level2name=N'ID_SR_PHYS_PERSON_EXEX',
@value=N'ID Физическое лицо'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTERNAL_EXECUTOR_EXEX',
@level2name=N'ID_R_REASON_OUTSOURCING_EXEX',
@value=N'ID Причина привлечения стороннего специалиста'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTERNAL_EXECUTOR_EXEX',
@level2name=N'COMMON_DESCR_EXEX',
@value=N'Общее описание задачи'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTERNAL_EXECUTOR_EXEX',
@level2name=N'COMMENT_EXEX',
@value=N'Комментарий'

declare @id bigint;
use <ИМЯ БД>;
if exists (select 'x' from sysobjects obj where obj.name = 'EXTERNAL_EXECUTOR_EXEX' and
obj.type = 'U')
    select @id = ISNULL(MAX(t.ID_EXTERNAL_EXECUTOR_EXEX), 0) from EXTERNAL_EXECUTOR_EXEX t;
else
    set @id = 1;
use PM_INSTALL_S_DATA;
if exists(select 'x' from S_TB_SEQ seq where seq.SEQ_NAME = 'SEQ_EXTERNAL_EXECUTOR_EXEX')
    update S_TB_SEQ set SEQ_VALUE = @id, INCREMENT_BY = 1
    where SEQ_NAME = 'SEQ_EXTERNAL_EXECUTOR_EXEX';
else
    insert S_TB_SEQ (SEQ_NAME, INCREMENT_BY, SEQ_VALUE)
    values ('SEQ_EXTERNAL_EXECUTOR_EXEX', 1, @id);

go

EXTER_EXEC_LIST_JOB_POSI
USE [<ИМЯ БД>]
GO

/***** Object: Table [dbo].[EXTER_EXEC_LIST_JOB_POSI]    Script Date: 23.05.2018 16:19:51
*****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI](

```

```

[ID_EXTER_EXEC_LIST_JOB_POSI] [int] NOT NULL,
[ID_EXTERNAL_EXECUTOR_POSI] [numeric](8, 0) NOT NULL,
[NUMBER_PP_POSI] [numeric](3, 0) NOT NULL,
[DESCR_JOB_POSI] [varchar](20) NOT NULL,
[ID_SR_R_PROMPTNESS_POSI] [int] NULL,
[DATE_JOB_POSI] [datetime] NULL,
[VOL_POSI] [numeric](10, 3) NULL,
[ID_S_EI_POSI] [numeric](10, 0) NULL,
[RES_POSI] [varchar](200) NULL,
[DATE_INPUT_POSI] [datetime] NOT NULL,
[DATE_CHANGE_POSI] [datetime] NULL,
CONSTRAINT [XPKE_XTER_EXEC_LIST_JOB_POSI] PRIMARY KEY NONCLUSTERED
(
    [ID_EXTER_EXEC_LIST_JOB_POSI] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [XAK1EXTER_EXEC_LIST_JOB_POSI] UNIQUE NONCLUSTERED
(
    [ID_EXTERNAL_EXECUTOR_POSI] ASC,
    [NUMBER_PP_POSI] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI] ADD DEFAULT (getdate()) FOR [DATE_INPUT_POSI]
GO

ALTER TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI] WITH CHECK ADD CONSTRAINT [R_2703] FOREIGN
KEY([ID_EXTERNAL_EXECUTOR_POSI])
REFERENCES [dbo].[EXTERNAL_EXECUTOR_EXEX] ([ID_EXTERNAL_EXECUTOR_EXEX])
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI] CHECK CONSTRAINT [R_2703]
GO

ALTER TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI] WITH CHECK ADD CONSTRAINT [R_2855] FOREIGN
KEY([ID_S_EI_POSI])
REFERENCES [dbo].[S_EI_EIEI] ([ID_S_EI_EIEI])
GO

ALTER TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI] CHECK CONSTRAINT [R_2855]
GO

ALTER TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI] WITH CHECK ADD CONSTRAINT [R_2856] FOREIGN
KEY([ID_SR_R_PROMPTNESS_POSI])
REFERENCES [dbo].[SR_R_PROMPTNESS_SRPR] ([ID_SR_R_PROMPTNESS_SRPR])
GO

ALTER TABLE [dbo].[EXTER_EXEC_LIST_JOB_POSI] CHECK CONSTRAINT [R_2856]
GO

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SHEMA', @level0name=N'dbo', @level1type=N'TABLE',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@value=N'СПИСОК РАБОТ ИСПОЛНИТЕЛЯ'
```

```

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'ID_EXTER_EXEC_LIST_JOB_POSI',
@value=N'ID Список работ исполнителя'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'ID_EXTERNAL_EXECUTOR_POSI',
@value=N'ID Сторонний исполнитель'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'NUMBER_PP_POSI',
@value=N'Номер пп'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'DESCR_JOB_POSI',
@value=N'Описание работ'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'ID_SR_R_PROMPTNESS_POSI',
@value=N'ID Очередность работ'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'DATE_JOB_POSI',
@value=N'Дата работ'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'VOL_POSI',
@value=N'Объем работ'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'ID_S_EI_POSI',
@value=N'ID ЕИ объема работ'

EXEC sys.sp_addextendedproperty @name=N'Comment',
@level0type=N'SCHEMA', @level0name=N'dbo', @level1type=N'TABLE', @level2type=N'COLUMN',
@level1name=N'EXTER_EXEC_LIST_JOB_POSI',
@level2name=N'RES_POSI',
@value=N'Результат работ'

declare @id bigint;
use <ИМЯ БД>;
if exists (select 'x' from sysobjects obj where obj.name = 'EXTER_EXEC_LIST_JOB_POSI' and
obj.type = 'U')
    select @id = ISNULL(MAX(t.ID_EXTER_EXEC_LIST_JOB_POSI), 0) from EXTER_EXEC_LIST_JOB_POSI
t;
else
    set @id = 1;

```

```

use PM_INSTALL_S_DATA;
if exists(select 'x' from S_TB_SEQ seq where seq.SEQ_NAME = 'SEQ_EXTER_EXEC_LIST_JOB_POSI')
    update S_TB_SEQ set SEQ_VALUE = @id, INCREMENT_BY = 1
    where SEQ_NAME = 'SEQ_EXTER_EXEC_LIST_JOB_POSI';
else
    insert S_TB_SEQ (SEQ_NAME, INCREMENT_BY, SEQ_VALUE)
    values ('SEQ_EXTER_EXEC_LIST_JOB_POSI', 1, @id);

go

```

## СУБД ORACLE

Перед запуском скриптов нужно указать имя БД вместо текста <Имя БД>.

### R\_REASON\_OUTSOURCING

```

CREATE TABLE <ИМЯ БД>.R_REASON_OUTSOURCING
(
    ID_R_REASON_OUTSOURCING    NUMBER(3)           NOT NULL,
    NUMBER_PP                  NUMBER(3)           NOT NULL,
    NAME_REASON                 VARCHAR2(64 BYTE)   NOT NULL,
    VVOD_ID_CONTRACTOR         VARCHAR2(30 BYTE)   NOT NULL,
    CHANGE_ID_CONTRACTOR       VARCHAR2(30 BYTE),
    DATE_INPUT                 DATE                DEFAULT sysdate          NOT NULL,
    DATE_CHANGE                DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;

COMMENT ON TABLE <ИМЯ БД>.R_REASON_OUTSOURCING IS 'С_ПРИЧИНА ПРИВЛЕЧЕНИЯ СТОРОННЕГО СПЕЦИАЛИСТА';

COMMENT ON COLUMN <ИМЯ БД>.R_REASON_OUTSOURCING.ID_R_REASON_OUTSOURCING IS 'Id Причина привлечения стороннего специалиста';

COMMENT ON COLUMN <ИМЯ БД>.R_REASON_OUTSOURCING.NUMBER_PP IS 'Номер пп';

COMMENT ON COLUMN <ИМЯ БД>.R_REASON_OUTSOURCING.NAME_REASON IS 'Наименование причины';

COMMENT ON COLUMN <ИМЯ БД>.R_REASON_OUTSOURCING.VVOD_ID_CONTRACTOR IS 'Автор ввода';

COMMENT ON COLUMN <ИМЯ БД>.R_REASON_OUTSOURCING.CHANGE_ID_CONTRACTOR IS 'Автор изменения';

COMMENT ON COLUMN <ИМЯ БД>.R_REASON_OUTSOURCING.DATE_INPUT IS 'Дата ввода';

COMMENT ON COLUMN <ИМЯ БД>.R_REASON_OUTSOURCING.DATE_CHANGE IS 'Дата изменения';

CREATE INDEX <ИМЯ БД>.XAK1R_REASON_OUTSOURCING ON <ИМЯ БД>.R_REASON_OUTSOURCING
(NUMBER_PP)
LOGGING
NOPARALLEL;

CREATE INDEX <ИМЯ БД>.XAK2R_REASON_OUTSOURCING ON <ИМЯ БД>.R_REASON_OUTSOURCING
(NAME_REASON)
LOGGING
NOPARALLEL;

CREATE UNIQUE INDEX <ИМЯ БД>.XPKR_REASON_OUTSOURCING ON <ИМЯ БД>.R_REASON_OUTSOURCING
(ID_R_REASON_OUTSOURCING)

```



```
COMMENT ON COLUMN <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX.ID_SR_PHYS_PERSON_EXEX IS 'ID_Физическое
лицо';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX.ID_R_REASON_OUTSOURCING_EXEX IS 'Id
Причина привлечения стороннего специалиста';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX.COMMON_DESCR_EXEX IS 'Общее описание
задачи';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX.COMMENT_EXEX IS 'Комментарий';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX.DATE_INPUT_EXEX IS 'Дата ввода';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX.DATE_CHANGE_EXEX IS 'Дата изменения';
```

```
CREATE INDEX <ИМЯ БД>.XAK1EXTERNAL_EXECUTOR_EXEX ON <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX
(ID_SR_R_PROJECT_EXEX, ID_SR_PHYS_PERSON_EXEX)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX <ИМЯ БД>.XIF1442EXTERNAL_EXECUTOR_EXEX ON <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX
(ID_R_REASON_OUTSOURCING_EXEX)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX <ИМЯ БД>.XIF1443EXTERNAL_EXECUTOR_EXEX ON <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX
(ID_SR_R_PROJECT_EXEX)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX <ИМЯ БД>.XIF1444EXTERNAL_EXECUTOR_EXEX ON <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX
(ID_SR_PHYS_PERSON_EXEX)
LOGGING
NOPARALLEL;
```

```
CREATE UNIQUE INDEX <ИМЯ БД>.XPKEXTERNAL_EXECUTOR_EXEX ON <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX
(ID_EXTERNAL_EXECUTOR_EXEX)
LOGGING
NOPARALLEL;
```

```
ALTER TABLE <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX ADD (
  CONSTRAINT EXTERNAL_EXECUTOR_EXEXV_DN
  CHECK (COMMENT_EXEX IN ('Д', 'Н'))
  ENABLE VALIDATE,
  CONSTRAINT XPKEXTERNAL_EXECUTOR_EXEX
  PRIMARY KEY
  (ID_EXTERNAL_EXECUTOR_EXEX)
  USING INDEX <ИМЯ БД>.XPKEXTERNAL_EXECUTOR_EXEX
  ENABLE VALIDATE,
  CONSTRAINT XAK1EXTERNAL_EXECUTOR_EXEX
  UNIQUE (ID_SR_R_PROJECT_EXEX, ID_SR_PHYS_PERSON_EXEX)
  DEFERRABLE INITIALLY IMMEDIATE
  USING INDEX <ИМЯ БД>.XAK1EXTERNAL_EXECUTOR_EXEX
  ENABLE VALIDATE);
```

```
ALTER TABLE <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX ADD (
  CONSTRAINT R_2576
  FOREIGN KEY (ID_SR_R_PROJECT_EXEX)
  REFERENCES <ИМЯ БД>.SR_R_PROJECT_SRPJ (ID_SR_R_PROJECT_SRPJ)
  DEFERRABLE INITIALLY IMMEDIATE
  ENABLE VALIDATE,
```



```
COMMENT ON COLUMN <ИМЯ БД>.EXTER EXEC LIST JOB POSI.DESCR JOB POSI IS 'Описание работ';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI.ID_SR_R_PROMPTNESS_POSI IS 'Очередность работ';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI.DATE_JOB_POSI IS 'Дата работ';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI.VOL_POSI IS 'Объем работ';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI.ID_S_EI_POSI IS 'ЕИ работ';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI.RES_POSI IS 'Результат работ';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI.DATE_INPUT_POSI IS 'Дата ввода';
```

```
COMMENT ON COLUMN <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI.DATE_CHANGE_POSI IS 'Дата изменения';
```

```
CREATE INDEX <ИМЯ БД>.XAK1EXTER_EXEC_LIST_JOB_POSI ON <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI
(ID_EXTERNAL_EXECUTOR_POSI, NUMBER_PP_POSI)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX <ИМЯ БД>.XIF1457EXTER_EXEC_LIST_JOB_POS ON <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI
(ID_EXTERNAL_EXECUTOR_POSI)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX <ИМЯ БД>.XIF1458EXTER_EXEC_LIST_JOB_POS ON <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI
(ID_S_EI_POSI)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX <ИМЯ БД>.XIF1459EXTER_EXEC_LIST_JOB_POS ON <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI
(ID_SR_R_PROMPTNESS_POSI)
LOGGING
NOPARALLEL;
```

```
CREATE UNIQUE INDEX <ИМЯ БД>.XPKEEXTER_EXEC_LIST_JOB_POSI ON <ИМЯ
БД>.EXTER_EXEC_LIST_JOB_POSI
(ID_EXTER_EXEC_LIST_JOB_POSI)
LOGGING
NOPARALLEL;
```

```
ALTER TABLE <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI ADD (
CONSTRAINT XPKEEXTER_EXEC_LIST_JOB_POSI
PRIMARY KEY
(ID_EXTER_EXEC_LIST_JOB_POSI)
USING INDEX <ИМЯ БД>.XPKEEXTER_EXEC_LIST_JOB_POSI
ENABLE VALIDATE,
CONSTRAINT XAK1EXTER_EXEC_LIST_JOB_POSI
UNIQUE (ID_EXTERNAL_EXECUTOR_POSI, NUMBER_PP_POSI)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX <ИМЯ БД>.XAK1EXTER_EXEC_LIST_JOB_POSI
ENABLE VALIDATE);
```

```
ALTER TABLE <ИМЯ БД>.EXTER_EXEC_LIST_JOB_POSI ADD (
CONSTRAINT R_2578
FOREIGN KEY (ID_SR_R_PROMPTNESS_POSI)
REFERENCES <ИМЯ БД>.SR_R_PROMPTNESS_SRPR (ID_SR_R_PROMPTNESS_SRPR)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE,
CONSTRAINT R_2703
FOREIGN KEY (ID_EXTERNAL_EXECUTOR_POSI)
REFERENCES <ИМЯ БД>.EXTERNAL_EXECUTOR_EXEX (ID_EXTERNAL_EXECUTOR_EXEX)
```

[illegible]