

Группа компаний "Информация и управление"

*Инструментальные средства объектной
разработки программных систем
S_Технология (SDK S_Технология)*

Порядок создания приложения Windows Forms. Состав проекта

Редакция 2.0.0.X

Воронеж, 2018

АННОТАЦИЯ

Документ предназначен для программиста, работающего с пакетом SDK *S_Технология*. В документе подробно описано применение инструментальных средств SDK *S_Технология* и расширения для MS Visual Studio – создание нового проекта, модулей таблиц, классов Forms для *S_Технологии* при помощи конструктора форм и другие возможности. Описание приведено для архитектуры Windows Forms.

Документ по шагам описывает весь процесс создания нового проекта.

На первом шаге (раздел **Шаг 1**) с помощью специального инструмента-расширения VS «**IMWinFormsProject**» создается VS-проект с поддержкой *S_Технологии*. В процессе создания VS-проекта вызывается особое приложение – **Конфигуратор**. Он позволяет автоматически создать и (или) настроить необходимые для *S_Технологии* БД, таблицы, пользователей и другие объекты БД. Это же приложение заполняет таблицы необходимыми данными. Если необходимые БД/таблицы уже были созданы раньше, выполняется только проверка их корректности.

Следующим этапом будет создание **модулей таблиц**. Это специальные модули, содержащие два класса, потомки классов – `s_DataObject` и `s_EntityObject`. Совокупность этих двух классов позволяет обращаться к таблице, добавлять/изменять/удалять записи и многое другое. Классы описания таблиц являются основой применения *S_Технологии*. В разделе **Шаг 2** описан процесс автоматического создания модулей таблиц на примере ряда демонстрационных таблиц, включенных в пакет.

В разделе **Шаг 3** показана простейшая визуализация таблицы, которая выполняется с помощью механизма *S_Технологии* – «Справочники». Любые таблицы БД можно показать в стандартной форме только с помощью **модуля таблицы** без создания каких-то дополнительных форм. Стандартная форма имеет ограниченный функционал, но достаточно широкий. Она поддерживает добавление/изменение/удаление записи и возможность работы с записью в двух видах – Список/Карточка. Приведено подробное описание стандартной формы.

В разделе **Шаг 4** показан процесс создания сложных форм – **Графов**. Для их создания используется инструмент «**Конструктор форм**». С его помощью на базе **модулей таблиц** можно создавать формы практически любой сложности. Форма-граф может включать в себя любое разумное количество форм, отображающих разные таблицы, связанных друг с другом и имеющих разный вид/поведение.

Раздел **Шаг 5** описывает способы подключения формы-графа к проекту.

Инструментом «**Конструктор форм**» можно не только создавать, но редактировать имеющийся «**Граф**». Например, добавить отображение таблицы в специальной форме. При этом будет возможность расширить функционал для данной таблицы, добавив код непосредственно в класс формы. Этот процесс описан в разделе **Шаг 6**.

В разделе **Шаг 7** очень кратко описано использование инструмента «**Конструктор запросов**». Основное описание инструмента «Конструктор запросов» дано в документе “IM Конструктор запросов”.

Шаг 8 описывает процесс переименования проекта/приложения, созданного в рамках *S_Технологии* инструментом-расширением «**IMWinFormsProject**». Там же приведен краткий сценарий дальнейшего использования проекта.

В Приложении 1 приведены примеры применения Генератора форм.

Инструментальные средства объектной разработки программных систем *S_Технология* (SDK *S_Технология*) разработаны группой компаний "Информация и управление" (компания "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

ВВЕДЕНИЕ	5
ШАГ 1. СОЗДАНИЕ НОВОГО ПРОЕКТА.....	5
ШАГ 2. НАПОЛНЕНИЕ ПРОЕКТА МОДУЛЯМИ ТАБЛИЦ НА ПРИМЕРЕ НЕСКОЛЬКИХ ДЕМОНСТРАЦИОННЫХ ТАБЛИЦ.....	13
ШАГ 3. ИСПОЛЬЗОВАНИЕ ТАБЛИЦЫ В КАЧЕСТВЕ СПРАВОЧНИКА	23
ШАГ 4. СОЗДАНИЕ ФОРМЫ ПРИ ПОМОЩИ КОНСТРУКТОРА ФОРМ.....	30
ШАГ 5. ПОДКЛЮЧЕНИЕ ФОРМЫ-ГРАФА К ПРОЕКТУ	38
СЦЕНАРИЙ 1. АВТОМАТИЧЕСКОЕ ДОБАВЛЕНИЕ ГРАФА К ПРОЕКТУ	38
СЦЕНАРИЙ 2. ВЫЗОВ ФОРМЫ-ГРАФА	39
ШАГ 6. РЕДАКТИРОВАНИЕ ГРАФА. ДОБАВЛЕНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ ФОРМЫ И КНОПКИ НА НЕЙ	43
1.1. СВОЙСТВА ГРАФА	52
1.2. СВОЙСТВА УЗЛА	54
1.3. СВОЙСТВА СВЯЗИ.....	57
ШАГ 7. ИНСТРУМЕНТ «КОНСТРУКТОР ЗАПРОСОВ»	58
ШАГ 8. СОЗДАНИЕ НОВОГО ПРОЕКТА И ЕГО ПЕРЕИМЕНОВАНИЕ. ДАЛЬНЕЙШЕЕ РАЗВИТИЕ ПРИЛОЖЕНИЯ	60
СОЗДАНИЕ ПРОЕКТА С ПРОИЗВОЛЬНЫМ ИМЕНЕМ СБОРКИ.....	60
ПОЛЬЗОВАТЕЛЬ ПРОЕКТА. СИСТЕМА ПРИВИЛЕГИЙ	65
РАЗВИТИЕ ПРИЛОЖЕНИЯ	68
ОБЩИЙ СЦЕНАРИЙ ИСПОЛЬЗОВАНИЯ ПРОЕКТА	68
ПРИЛОЖЕНИЕ 1. ПРИМЕРЫ ПРИМЕНЕНИЯ ГЕНЕРАТОРА ФОРМ	69
ФОРМЫ, ГРАФЫ	69
<i>Как перекрыть обработчик базовой кнопки формы узла (показ графа-карточки).....</i>	<i>69</i>
<i>Как перекрыть обработчик базовой кнопки формы графа</i>	<i>69</i>
<i>Как создать фильтр со списком значений статичный (custom)</i>	<i>70</i>
<i>Как создать фильтр со списком значений заполняемых в runtime</i>	<i>71</i>
<i>Как подписаться на обработку всех графов</i>	<i>72</i>
<i>Как делать ручную карточку (привязка своих контролов к атрибутам выборки)</i>	<i>72</i>
<i>Как подписаться на изменение узла графа</i>	<i>73</i>
<i>Как использовать форму с прогрессом (простая)</i>	<i>73</i>
<i>Как использовать форму с прогрессом (с этапами, как в импорте)</i>	<i>73</i>
<i>Как расположить фильтры в несколько рядов (задание каждому отбору своего места)</i>	<i>74</i>
<i>Как использовать базовые панели формы для размещения своих элементов (мастер->деталь->деталь) ..</i>	<i>75</i>
<i>Как задать доп. условие на узел</i>	<i>76</i>
<i>Как по кнопке сменить выборку (смена selectionName).....</i>	<i>76</i>
<i>Как добавить кнопку на базовую панель формы (через функции)</i>	<i>77</i>
<i>Как управлять доступностью кнопок в форме узла</i>	<i>77</i>
<i>Как назначить картинку для колонки и обработать клик по ней.....</i>	<i>77</i>
СУЩНОСТИ	78
<i>Как добавить бизнес-логику в сущность</i>	<i>78</i>
<i>Как задать сортировку дата-класса</i>	<i>78</i>
<i>Как назначить для сущности показ графа (перекрытие метода ShowData)</i>	<i>78</i>
<i>Как передать доп. условие или параметры в вызываемую по ссылочной кнопке сущность</i>	<i>79</i>
<i>Как определить набор полей для выборки.....</i>	<i>79</i>
<i>Как создать вычисляемое поле.....</i>	<i>80</i>
<i>Как обработать изменение атрибута</i>	<i>81</i>
<i>Как добавить ссылочное поле и наложить доп. условие на него</i>	<i>81</i>
ПРИЛОЖЕНИЕ 2. СОСТАВ И НАЗНАЧЕНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ В НОВОМ ПРОЕКТЕ.....	84
ПРИЛОЖЕНИЕ 3. СОСТАВ И НАЗНАЧЕНИЕ ПРИВИЛЕГИЙ НА ОПЕРАЦИИ.....	86

ПРИЛОЖЕНИЕ 4. СКРИПТЫ НА СОЗДАНИЕ ДЕМОНСТРАЦИОННЫХ ТАБЛИЦ ДЛЯ MS SQL SERVER И ORACLE	89
Скрипты для СОЗДАНИЕ ДЕМОНСТРАЦИОННЫХ ТАБЛИЦ для MS SQL SERVER	89
Скрипты для СОЗДАНИЕ ДЕМОНСТРАЦИОННЫХ ТАБЛИЦ для ORACLE	106

Введение

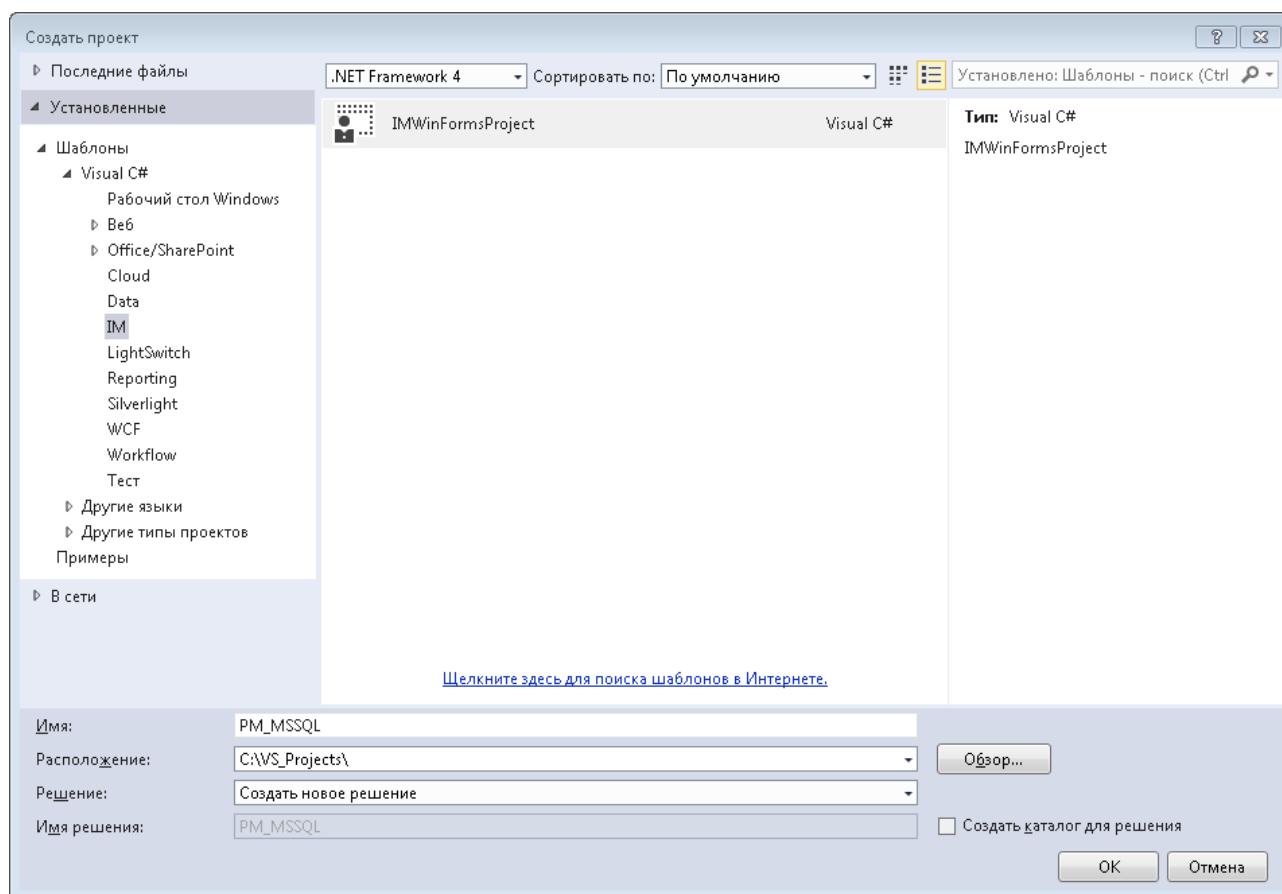
Инструментальные средства объектной разработки программных систем *S_Технология* предназначены для разработки информационных систем любой сложности на платформе .Net в среде разработки Microsoft Visual Studio на языке C# для архитектур Windows Forms и Web.

Шаг 1. Создание нового проекта

Внимание! Данный документ предполагает, что на компьютере установлены инструментальные средства SDK *S_Технология* и расширение для Microsoft Visual Studio. Порядок установки см. в отдельном документе «IM Установка SDK *S_Технология*».

Для создания нового проекта нужно запустить среду MS Visual Studio. В меню «Файл» - «Создать» - «Проект» выбрать пункт меню «IMWinFormsProject» (Visual C#\IM\). Примечание. Этого пункта не будет в меню если приложение для MS VS не установлено.

Имя проекта можно задавать произвольно, но важно помнить, что расширение «IMWinFormsProject» создаст проект, у которого будет фиксированное имя сборки (имя исполняемого файла) – “PM”. Это связано со спецификой *S_Технологии*. Подробнее об этом и о том, как создать проект *S_Технологии* с другим именем сборки, см. [Шаг 8. Создание нового проекта и его переименование. Дальнейшее развитие приложения.](#)



После задания имени нового проекта на экране появится окно **Конфигуратора базы данных *S_приложения***:

В **Конфигураторе БД** предусмотрена работа с двумя видами серверов БД – MSSQL и Oracle. Если проект на выбранном сервере, с заданными “Id проекта” и “Id экземпляра проекта” создается впервые, **Конфигуратор** создаст Базы данных (MSSQL) или Пользователей/Схемы (Oracle), необходимые проекту.

В результате работы **Конфигуратора** на сервере БД появятся:

- MSSQL. Две БД, прикладная и системная, с именами <“Id экземпляра проекта”>_data и <“Id экземпляра проекта”>_S_data.
- Oracle. Четыре Схемы/Пользователя – S_DBA, RTIME_DBA, PM_DBA и <Прокси-пользователь>. Схема PM_DBA содержит прикладные таблицы. Схема S_DBA содержит системные таблицы.

В **Конфигураторе** необходимо заполнить представленные поля. Для разного типа серверов состав и смысл поле будет немного отличаться, но в обоих случаях это будут понятные в рамках сервера термины. Рассмотрим более подробно способ заполнения на примере сервера БД типа MSSQL.

Для более простого заполнения полей, рекомендуется воспользоваться сохраненным экземпляром конфигурации. Для этого необходимо в нижней левой части формы выбрать из выпадающего списка “Экземпляры конфигурации” нужный экземпляр – “MSSQL” или “IM_ORA”. Выбираем “MSSQL”.

После выбора “Экземпляра конфигурации” поля окажутся частично заполнены. Дополнительно следует уточнить значения в следующих полях.

1. Имя сервера – выпадающий список “Имя сервера”.
2. Пароль пользователя sa для выбранного сервера – поле “Пароль”.
3. Имя прокси-пользователя – поле “Прокси пользователь”. Через это имя входа, в дальнейшем будут работать все пользователи проекта. По умолчанию “custom”.
4. ID проекта – поле “Id проекта”. Определяет имя создаваемого Проекта и Приложения. В данном документе создается Проект с именем “Project PM” и в нем Приложение с именем “PM”. Как создать проект с другим именем приложения см. [Шаг 8. Создание нового проекта и его переименование. Дальнейшее развитие приложения.](#)
5. Установить CheckBox “Windows Forms проект” (в новой версии “Конфигуратора” этот флаг будет соответствовать флагу “Создавать config-файл проекта”, т.е. будет установлен по умолчанию).
6. Имя экземпляра проекта – поле “Id экземпляра проекта”. По этому имени, для проекта будут созданы (если нет) две базы данных с именами: <значение поля>_DATA и <значение поля>_S_DATA.

Поле ниже CheckBox “Использовать скрипты объектов проекта” определяет скрипты для создания дополнительных объектов баз данных (таблицы, хранимые процедуры и пр.). После выбора “Экземпляра конфигурации” оно будет заполнено и при первом запуске его не надо менять. В дальнейшем, когда БД уже существует, использование этих скриптов можно отключить, но можно и оставить. При наличии объектов проекта их повторное создание не выполняется.

Дополнительно рекомендуется установить CheckBox “Запомнить пароль”. Сохраненный пароль в дальнейшем упростит работу с Конфигуратором. Хранится пароль в зашифрованном виде.

Project Configuration 3.2.1

Параметры экземпляра конфигурации

- Тип сервера: MSSQL
- Имя сервера: MSSQL-V1\SQL_2014
- Пользователь: sa
- Пароль: *****
☒ Запомнить пароль
- Прокси пользователь: custom
- Id проекта: PM
☒ Windows Forms проект
- Id экземпляра проекта: PM_FROM_INSTALLER
- Databases:
- ☒ Создать config-файл проекта
C:\Users\sshaposhnikov\AppData\Local\Temp\PM_FROM_INSTALLER\PM_FROM_INSTALLER.config
- ☒ Использовать скрипты объектов проекта
C:\Program Files\IM_Project\Scripts

Log

Сохранить в Notepad Сохранить в файле Скопировать в clipboard

Конфигурировать проект Скрипты Скрипты системных данных Конфигурировать БД Выход

Экземпляры конфигураций

Выбрать: MSSQL Сохранить как... Удалить...

Когда все необходимые поля заполнены станет активна кнопка “Конфигурировать БД”. Нажатие этой кнопки запустит конфигурирование БД – будет выполнен поиск на сервере MSSQL-V1\SQL_2014 баз данных с именами PM_FROM_INSTALLER_DATA и PM_FROM_INSTALLER_S_DATA. Если БД не будут найдены, конфигуратор предложит создать их.

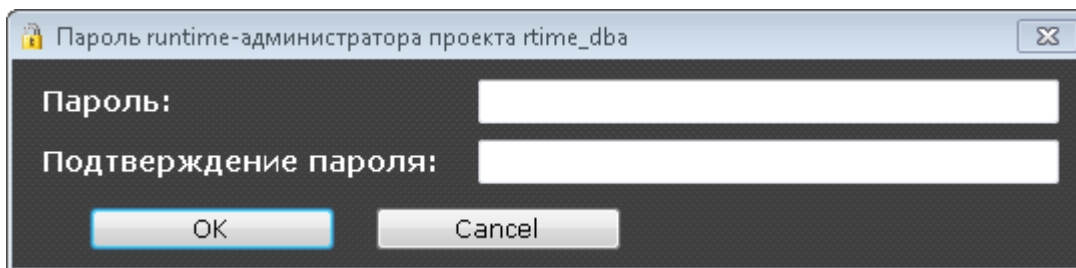
Сообщение

[s_pk001C] Не найдена база данных PM_FROM_INSTALLER_data для таблиц проекта. Будет создана база данных с именем PM_FROM_INSTALLER_data для таких таблиц.

Размер БД (МБ) 100

Продолжить Отказ

При отсутствии, на сервере также будут созданы специальные имена входа для пользователей-администраторов – s_dba и rtime_dba. Для них будет запрошен пароль.

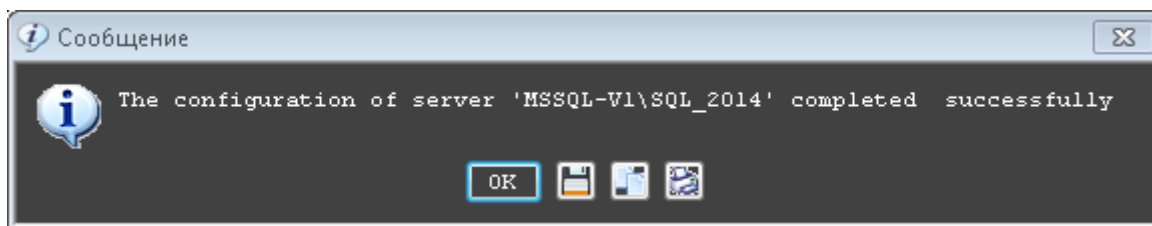


Кроме этого, при отсутствии, будет запрошен пароль и создано имя входа для прокси-пользователя custom (значение поля "Прокси пользователь").

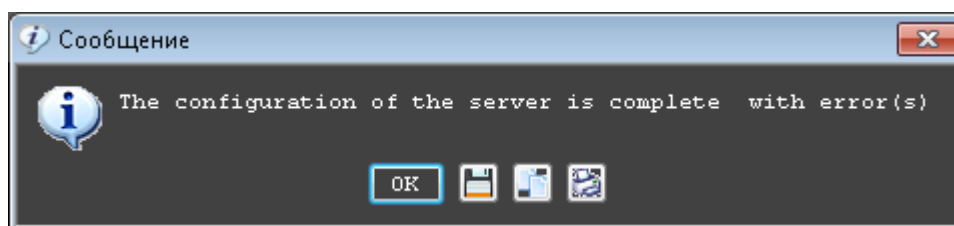


Возможно указание в качестве прокси-пользователя одного из уже существующих в базе пользователя, в этом случае нужно будет указать его пароль.

В случае успешного окончания работы конфигуратора появится сообщение об успешном завершении конфигурирования БД.

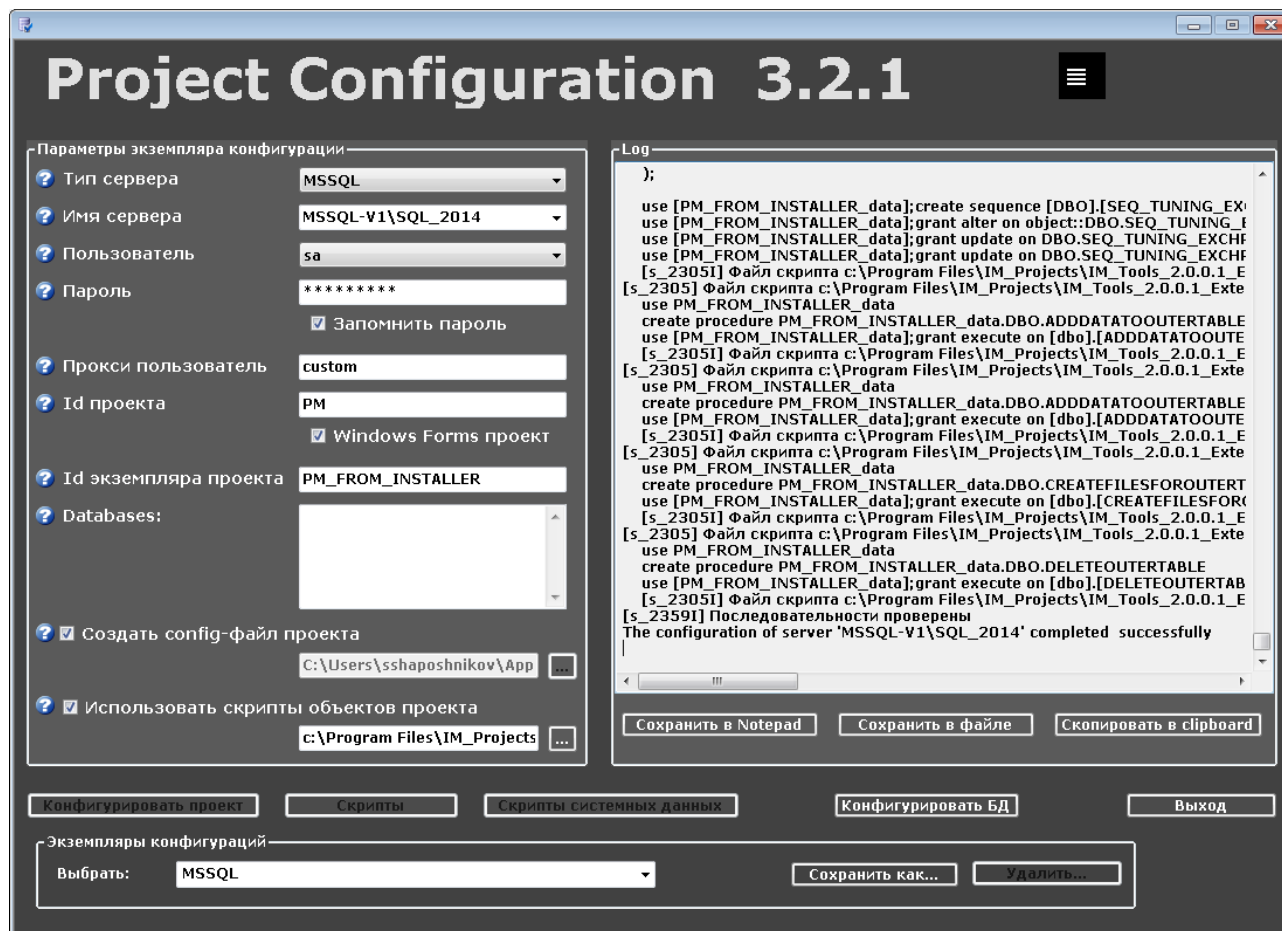


Или сообщение о некорректном завершении работы конфигуратора.



Это сообщение нужно закрыть по кнопке "OK".

Далее рекомендуется сохранить лог работы конфигуратора, для чего можно воспользоваться кнопкой "Сохранить в файле".



После сохранения лога можно закрыть Конфигуратор.

В результате конфигурирования БД были созданы (или обновлены) две базы данных с именами PM_FROM_INSTALLER_DATA и PM_FROM_INSTALLER_S_DATA.

Если базы данных создаются впервые, нужно выполнить дополнительные скрипты настройки сервера (см. описание в главе *Настройка инструментальных средств для различных СУБД* документа «IM Установка SDK S_Технология»).

База данных PM_FROM_INSTALLER_S_DATA содержит, в зависимости от версии MS SQL, 31 или 32 системные таблицы с префиксом S_TB_. Эти таблицы служат для обеспечения правильной работы S_Технологии, их нельзя корректировать и удалять. Желательно эту БД вообще не использовать для хранения других объектов.

База данных PM_FROM_INSTALLER_DATA содержит 21 прикладную таблицу, которые обеспечивают поддержку таких инструментов S_Технологии как отчёты, обмен данными, конструктор запросов и прочих (в зависимости от варианта поставки). Также в этой БД есть 8 прикладных таблиц, добавленных исключительно в целях демонстрации возможностей S_Технологии (см. ниже – [Шаг 2. Наполнение проекта модулями таблиц на примере нескольких демонстрационных таблиц.](#)). В эту БД можно добавлять новые объекты (таблицы) для дальнейшего развития проекта.

Список прикладных таблиц:

Специальные таблицы обеспечения S_Технологии:

- Меню проекта (PM_MENU)
- Журнал проекта (PM_LOG)

- Список представлений проекта (PM_VIEWS)

Общие системные справочники:

- ВЕРСИЯ БД (DB_VERSION_DBVR)
- ОПИСАНИЕ КЛАССА (CLASS_DESCRIPTION)
- СВОЙСТВО КЛАССА (OBJECTS_PROP_OBJP)
- Назначение единицы измерения (R_MEASURE_DS_MEDS)

Общие справочники пользователя:

- Единица измерения (S_EI_EIEI)
- Дополнительные данные объекта (DD_OBJECT_DDOB)

Инструмент “Конструктор запросов” :

- ЗАПРОС К БД (RQ_REQUEST_DB_RQDB)
- ЗАПРОС К БД ТЕКСТ (RQ_REQUEST_TXT_RQTX)

Инструмент “Обмен данными”.

Сист. справочники:

- ОБМЕН ДАННЫХ РАСПОЛОЖЕНИЕ (TUNING_DATASP_TEDP)
- ОБМЕН ДАННЫХ ФОРМАТ (TUNING_DATASF_TEDF)
- С_ТИП ВВОДА (TUNING_EXCH_TYPE_TETP)
- НАСТРОЙКА БД ИСТОЧНИКА (TUNING_TB_SOURCE_TTSR)
- НАСТРОЙКА БД ИСТОЧНИКА ПОЛЕ (TUNING_SOURCE_REL_TSRS)

Таблицы:

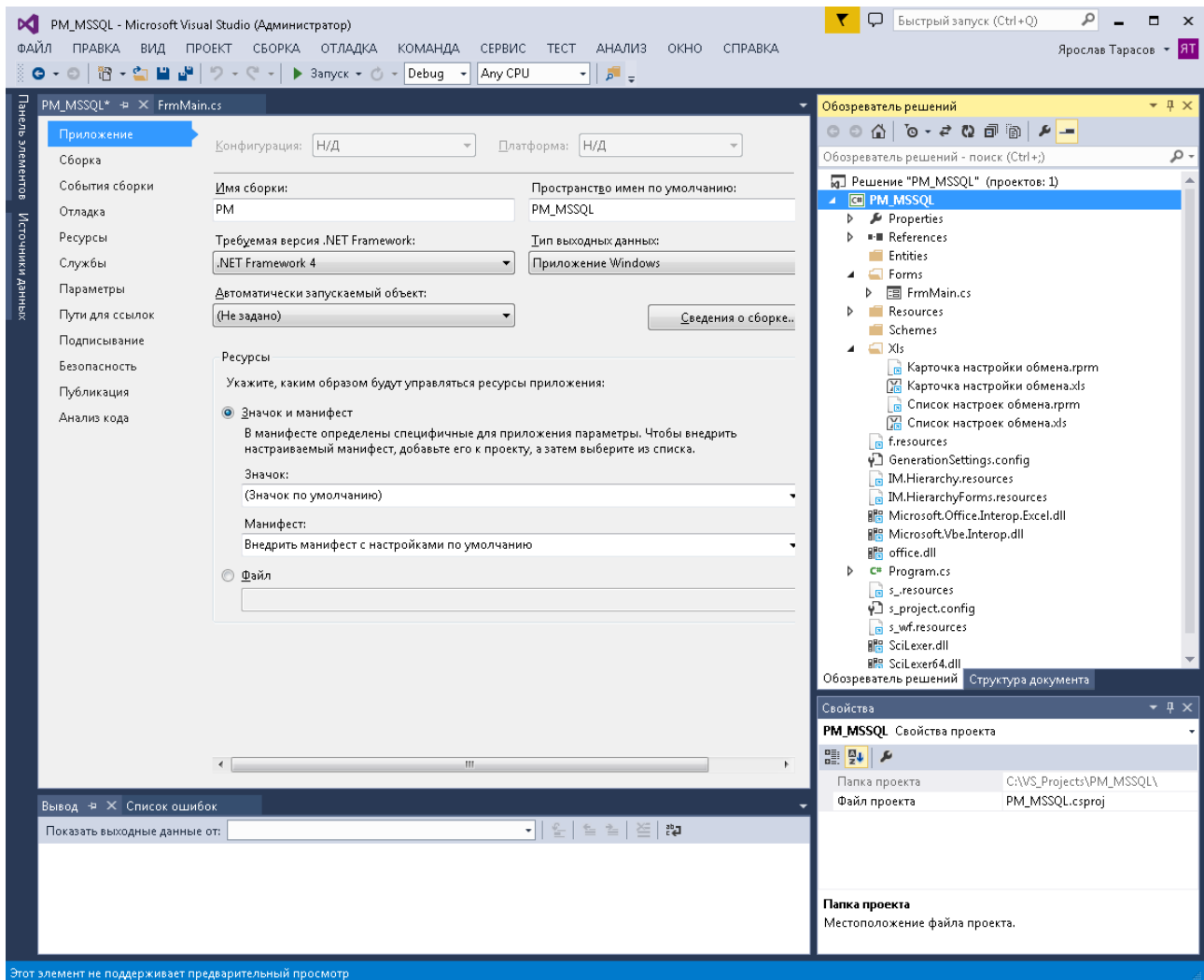
- НАСТРОЙКА ОБМЕНА (TUNING_EXCH_TEAA)
- НАСТРОЙКА ОБМЕНА ОБЪЕКТ (TUNING_EXCHOB_TEOB)
- НАСТРОЙКА ОБМЕНА СВОЙСТВА ОБЪЕКТА (TUNING_EXCHPR_TEPР)
- ИМПОРТ ВНЕШНЯЯ ТАБЛИЦА (IMPORT_EXEPTIONS_TEMP_IEXT)
- ИМПОРТ ВНЕШНЯЯ ТАБЛИЦА с BLOB полем (TEMP_EXT_TB_BLOB_TMPX)

В Приложении 2 приведен состав и назначение таблиц базы данных нового проекта.

Модули (см. ниже) для этих таблиц создавать не нужно, они уже существуют в соответствующих библиотеках инструментов.

Параметры подключения к базе данных сохраняются в файле s_project.config, который используется приложением при старте работы.

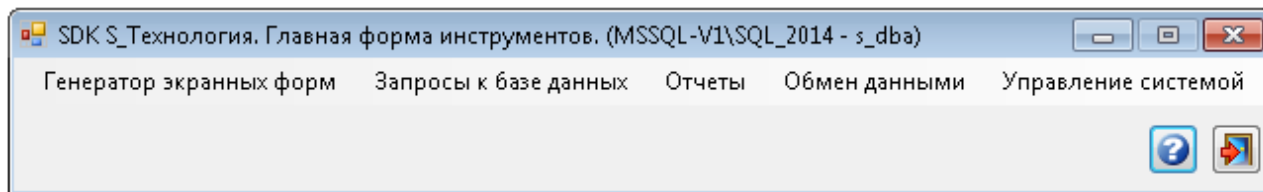
В результате создан IM-проект PM_MSSQL.csproj с именем сборки “PM”.



В созданном проекте существуют папки: **Forms** – для форм проекта, **Entities** – для модулей таблиц БД, **Schemes** – для xml-схем форм-графов.

Далее нужно запустить созданный проект на выполнение. Будет показана форма регистрации Пользователь/Пароль. Нужно ввести пользователя s_dba и пароль, который был назначен этому пользователю на этапе работы “Конфигуратора”.

После регистрации на экране появится главная форма проекта FrmMain.



Шаг 2. Наполнение проекта модулями таблиц на примере нескольких демонстрационных таблиц

В базе данных, созданной конфигуратором (PM_FROM_INSTALLER_DATA), будут присутствовать 8 демонстрационных таблиц с префиксом SR_:

1. SR_R_CLIENT_SRCL (ЗАКАЗЧИК)
2. SR_PHYS_PERSON_SRPP (ФИЗИЧЕСКОЕ ЛИЦО)
3. SR_R_PROJECT_SRPJ (ПРОЕКТ)
4. SR_R_STATUS_SRST (СТАТУС)
5. SR_R_EXECUTIVE_SREX (ИСПОЛНИТЕЛЬ)
6. SR_R_PROMPTNESS_SRPR (СРОЧНОСТЬ)
7. SR_REMARK_SRRM (ЗАМЕЧАНИЕ)
8. SR_DETAIL_REMARK_SRDR (ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ)

Ниже приведены диаграммы этих таблиц.

Диаграмма с наименованием атрибутов:

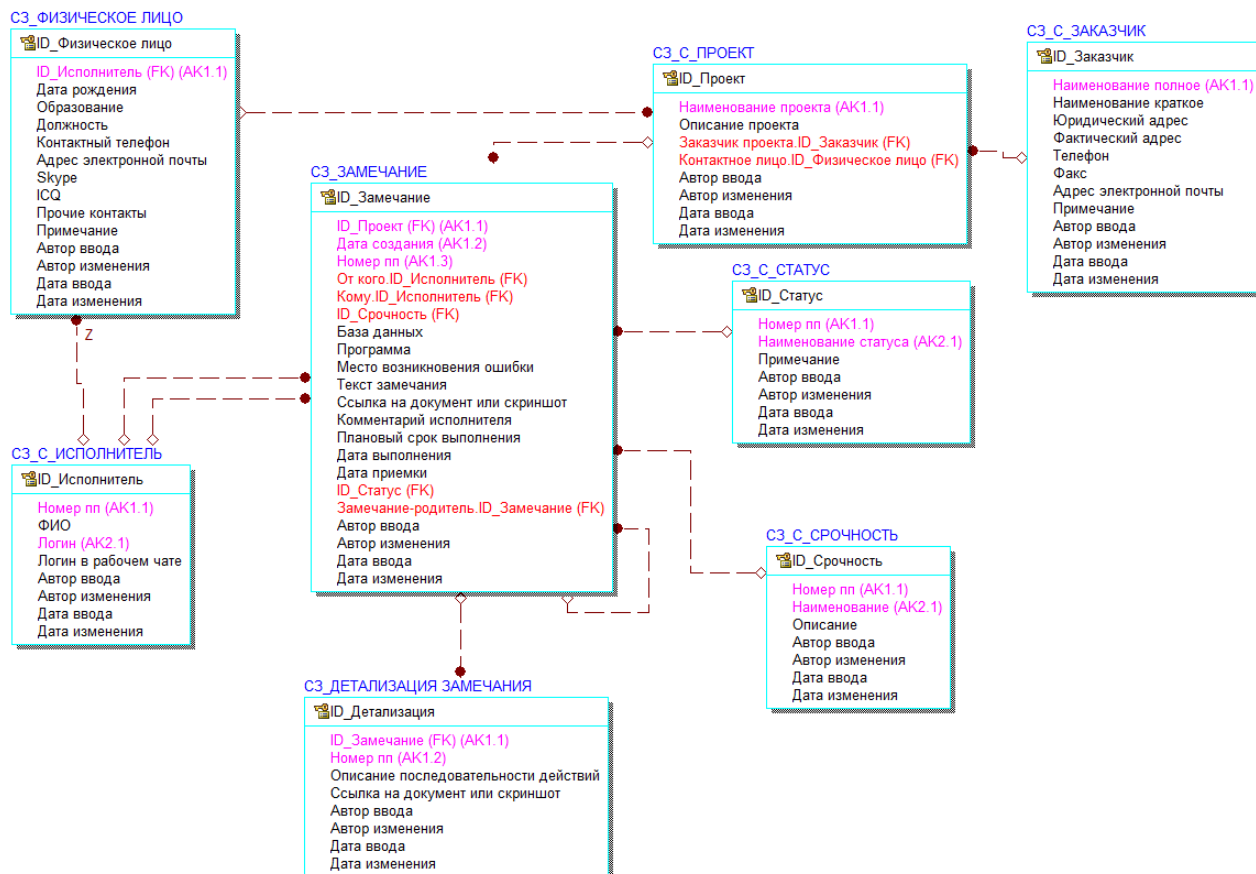
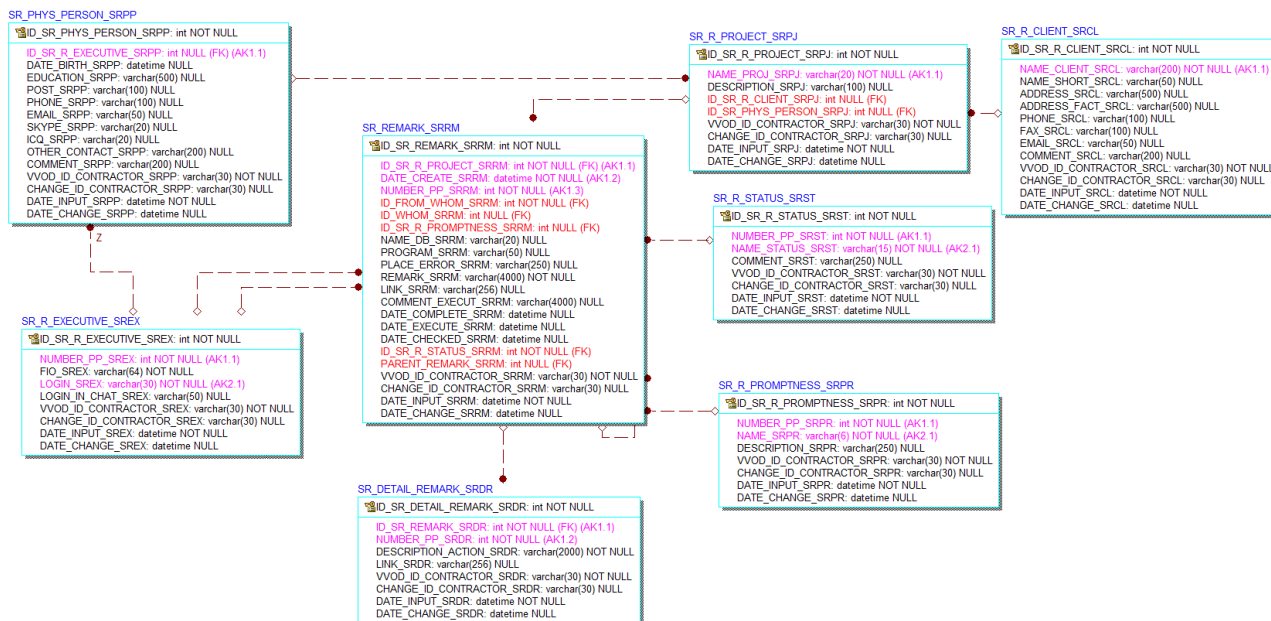


Диаграмма с именами полей таблиц:



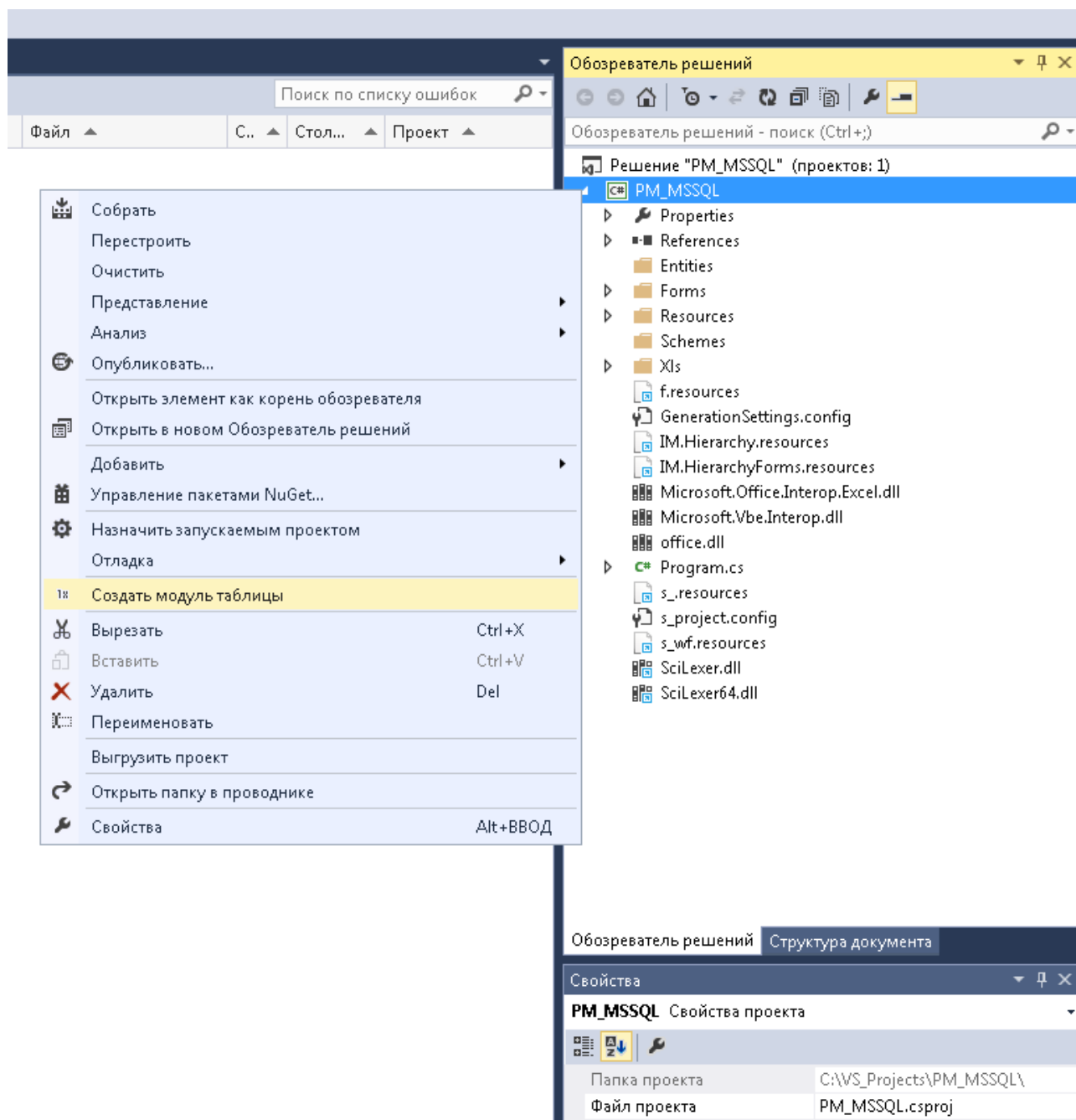
Эти таблицы являются частью действующего проекта “Project Manager” – PM, предназначенного для отслеживания выполнения заданий. Проект создан на базе S_Технологии.

Таблицы включены в поставку для опробования и наглядного изучения инструментов S_Технологии. Проекты Visual Studio для “Project Manager” для MS SQL Server и Oracle также включены в поставку в качестве примера. Кроме того, в **Приложении 4** приведены скрипты на создание этих таблиц для MS SQL Server и Oracle. Это сделано для того, чтобы продемонстрировать, как можно создать таблицы, чтобы сущности генерировались сразу с русскими именами таблиц и их полей; для этого необходимо, чтобы эти русские имена были сохранены в БД в комментариях к таблицам и полям соответственно. В противном случае русские имена придется прописывать в сущностях вручную.

Кроме таблиц, в базе созданы 8 последовательностей для генерации значений первичного ключа каждой таблицы. Их имена имеют вид – префикс SEQ_+<имя таблицы>.

Примечание. MSSQL-сервер версии ниже 11 (MSSQL2008 и ниже) еще не имеет механизма последовательностей. Для него используется механизм псевдо-последовательностей – будут внесены записи в специальную таблицу S_TB_SEQ.

Теперь требуется создать модули новых таблиц в новом проекте PM_MSSQL.csproj. Для того чтобы создать **модуль таблицы**, необходимо щелкнуть правой кнопкой мыши по заголовку проекта в обозревателе решений и выбрать пункт «Создать модуль таблицы».



Примечание. Создание модуля таблицы возможно только для проектов, которые имеют файл `s_project.config` (создаётся при успешном выполнении **шага 1**) и файл с настройками генерации `GenerationSettings.config`. Файл `GenerationSettings.config` создаётся шаблоном проекта и содержит информацию о параметрах создания модулей таблиц.

После выбора пункта меню создания модуля на экране появится окно мастера создания модулей таблиц.

The screenshot shows the 'Создание модуля таблицы' (Table Module Creation) window. The title bar includes a back arrow, a forward arrow, and a close button. The main area is titled 'Настройка соединения с базой' (Database Connection Setup). On the left, there is a sidebar with three items: 'Настройка соединения с базой...' (selected), 'Выбор таблиц' (Select Tables), and 'Завершение' (Completion). The main content area contains the following fields:

- Провайдер: MSSQL (dropdown)
- Сервер: MSSQL-VI\SQL_2014 (dropdown)
- Пользователь: s_dba (text input)
- Пароль: ***** (password input)
- ProjectInstanceId: PM_FROM_INSTALLER_DATA (dropdown)

At the bottom right, there are two buttons: 'Далее' (Next) and 'Отмена' (Cancel).

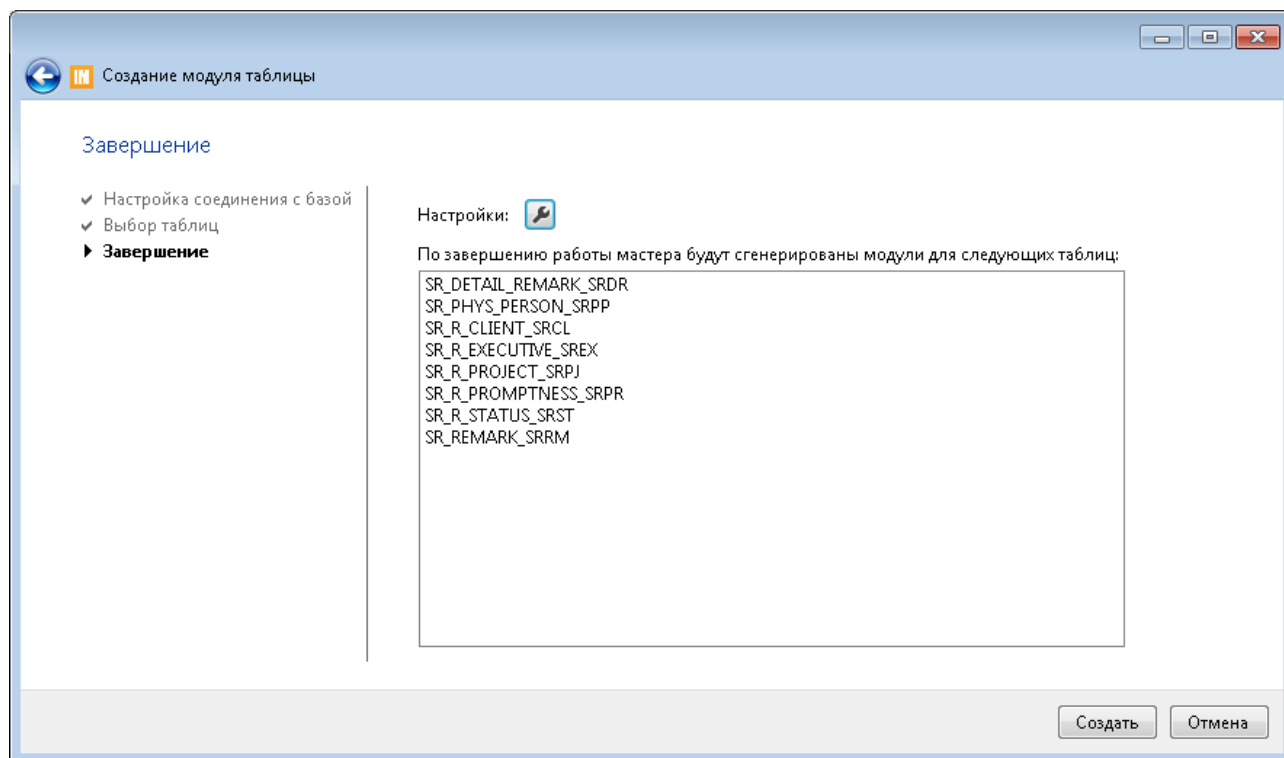
Здесь нужно ввести пароль пользователя s_dba, если он не введен, и нажать кнопку «Далее».

The screenshot shows the 'Создание модуля таблицы' (Table Module Creation) window. The title bar includes a back arrow, a forward arrow, and a close button. The main area is titled 'Выбор таблиц' (Select Tables). On the left, there is a sidebar with three items: 'Настройка соединения с базой' (selected), 'Выбор таблиц' (selected), and 'Завершение' (Completion). The main content area contains two lists:

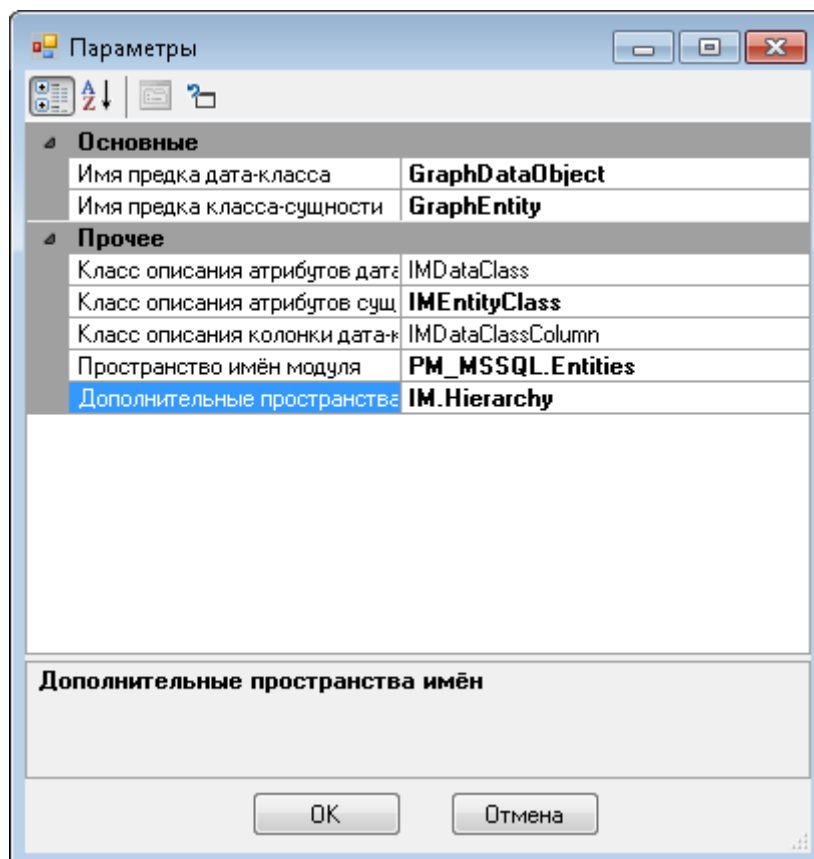
- Таблицы:** A list of tables with checkboxes. The checked tables are: SR_DETAIL_REMARK_SRDR, SR_PHYS_PERSON_SRPP, SR_R_CLIENT_SRCL, SR_R_EXECUTIVE_SREX, SR_R_PROJECT_SRPJ, SR_R_PROMPTNESS_SRPR, SR_R_STATUS_SRST, and SR_REMARK_SRRM.
- Столбцы:** A list of columns. The selected column is ID_SR_REMARK_SRRM.

At the bottom right, there are two buttons: 'Далее' (Next) and 'Отмена' (Cancel).

Следующим действием нужно выбрать таблицы (SR_R_CLIENT_SRCL, SR_PHYS_PERSON_SRPP, SR_R_PROJECT_SRPJ, SR_R_STATUS_SRST, SR_REMARK_SRRM, SR_DETAIL_REMARK_SRDR) и нажать кнопку «Далее».



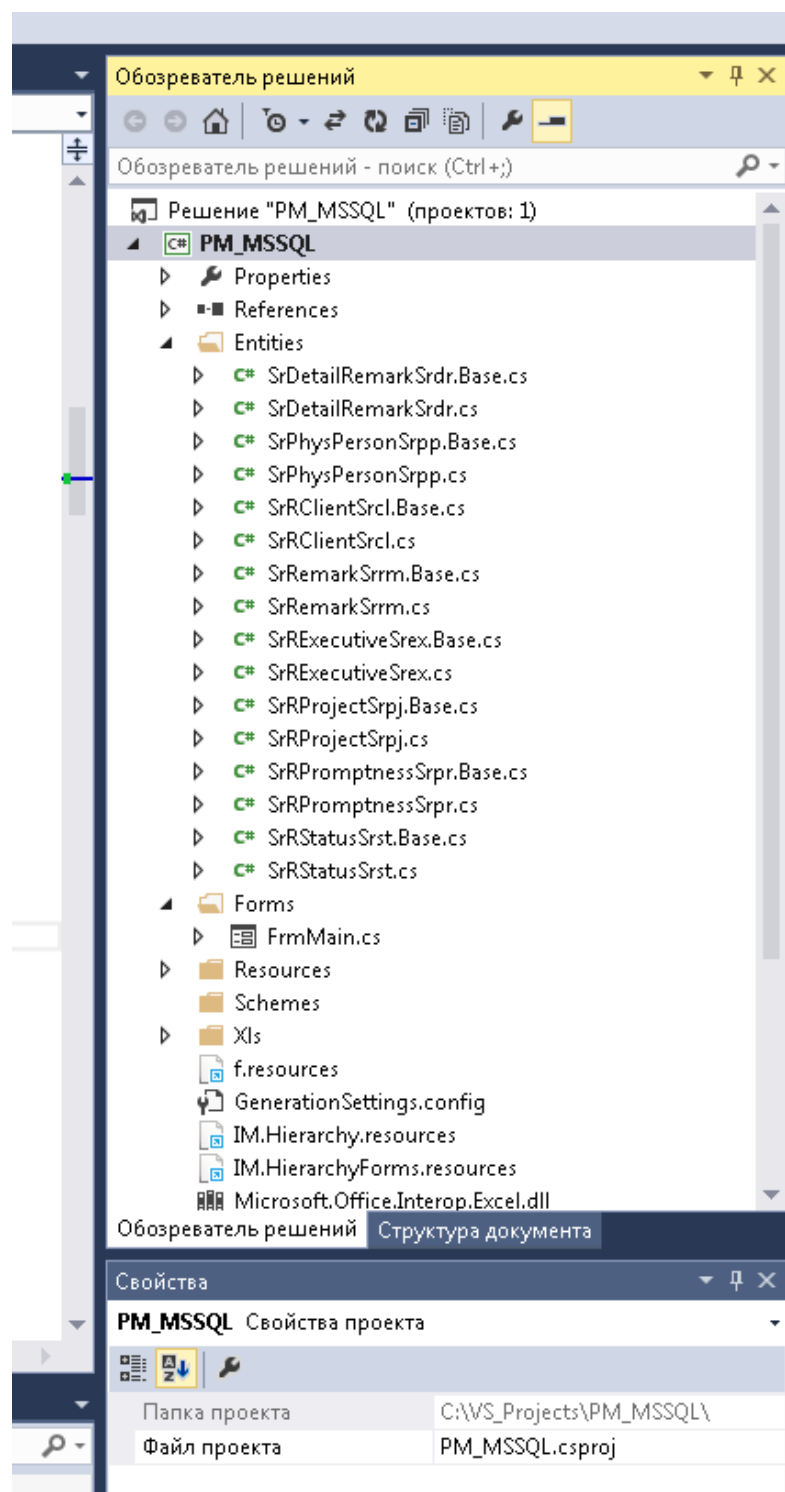
На завершающем этапе настроек мастера можно уточнить дополнительные параметры создания модулей таблиц, нажав кнопку «Настройки».



Изменённые значения параметров будут сохранены в файле `GenerationSettings.config` и использованы при повторных запусках плагина для создания модулей таблиц. Назначение и возможные значения параметров описаны в справке *S_.chm*, но на начальном этапе изучения *S_Технологии* менять их не рекомендуется. Кроме значения параметра «Пространство имен модуля» (можно поменять на любое другое название). Остальные параметры по умолчанию настроены правильно и, возможно, менять их никогда не придется.

Для создания модулей выбранных таблиц следует нажать кнопку «Создать».

После окончания работы мастера в проект (папка `Entities`) добавятся созданные модули таблиц (сущности). По два модуля на каждую таблицу.



Модули содержат два класса для каждой таблицы – Data-класс и Entity-класс. Подробно о Data и Entity классах таблиц см. в справке *S_.chm*.

Для простоты модули таблиц еще называются «Сущностями», по имени предка одного из классов таблицы «*s_EntityObject*».

Автоматически созданные сущности уже можно использовать для дальнейшей работы, не внося в них никаких изменений. Но есть некоторые тонкости. Самое главное, в модулях определены не все заголовки колонок – «Title» (определены только стандартные для *S_Технологии* «Автор ввода», «Автор изменения», «Дата ввода» и «Дата изменения»). Кроме того, не определено наименование таблицы на русском языке. Поэтому при визуализации сущности в заголовках будут отображаться имена полей на английском языке, что не всегда допустимо. Поэтому рекомендуется указать в сущности русское наименование таблицы и добавить свойство «Title» для тех колонок, в которых его нет. Ниже добавление русских имен в сущность показано на примере модуля таблицы СТАТУС (*SR_R_STATUS_SRST*) – *SrRStatusSrst.cs*.

Добавление русского наименования таблицы в атрибут *IMDataClass* – свойство *DefaultCaption*:

```
[IMDataClass(
    EntityClass = typeof(SrRStatusSrstEntity)
    , Sequence = @"SEQ_SR_R_STATUS_SRST"
    , LookupKeyColumns = @"Default"
    , LookupResultColumn = @"Default"
    , FormulaName = @"СТАТУС"
    , DefaultCaption = @"СТАТУС"
)]
```

Одновременно можно изменить значение свойства «FormulaName». «FormulaName» – имена полей для использования в формулах инструмента «Генератор отчетов», они должны соответствовать следующим правилам: используется только верхний регистр букв, пробелы не допускаются, заменяются знаком подчеркивания “_” (более подробно правила формирования имен «FormulaName» описаны в соответствующем руководстве по инструменту «Генератор отчетов»). В дальнейшем эти имена могут быть использованы в инструменте “Формулы/Отчеты”.

Следует обратить внимание на свойство **Sequence** (*Sequence = @"SEQ_SR_R_STATUS_SRST"*). Оно определяет имя последовательности, с помощью которой будет автоматически формироваться значение первичного ключа, при вставке новых записей в таблицу. Возможно, последовательность не нужна, например у таблицы нет первичного ключа или он имеет символьный тип. Тогда свойство **Sequence** надо удалить. В противном случае, **Sequence** с соответствующим именем (*SEQ_SR_R_STATUS_SRST*) должна присутствовать в БД (для MSSQL2011 и выше). Или должна быть запись в таблице *S_TB_SQL*, обеспечивающая псевдо-последовательность с таким именем (для MSSQL2008 и ниже).

Для демонстрационных таблиц нужные последовательности уже созданы, **но если будут создаваться новые таблицы с первичным ключом числового типа, нужно предусмотреть и создание последовательностей для них.**

Добавление/изменение значений свойств Title и FormulaName в атрибутах описания колонок (*#region Columns*):

Было –

```
[IMDataClassColumn(
    Name = SrRStatusSrstEntity.ColumnNumberPp
    , FormulaName = @"NUMBER_PP"
)]
```

Стало (одновременно изменено FormulaName) –

```
[IMDataClassColumn(
    Name = SrRStatusSrstEntity.ColumnNumberPp
    , Title = @"Номер пп"
    , FormulaName = @"НОМЕР_ПП"
)]
```

И так для всех `IMDataClassColumn`, которым требуются изменения.

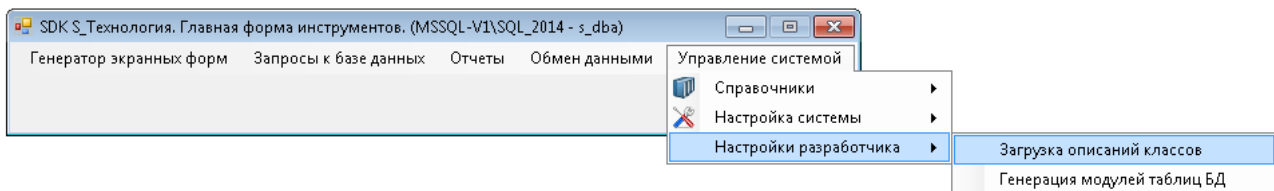
Добавление русского наименования таблицы в атрибут `IMEntityClass` – свойство `DefaultCaption` :

```
[IMEntityClass(
    DefaultCaption = @"СТАТУС"
    , DataClass = typeof(SrRStatusSrstData)
    , PrivilegeRead = @"Default"
    , PrivilegeWrite = @"Default"
)]
```

Таким образом, были созданы и отредактированы (дополнены русскими именами) модули таблиц (сущности). Чтобы сделать эти сущности, а следовательно и таблицы БД, доступными любому инструменту *S_Технологии*, требуется еще одно действие. Необходимо загрузить описание созданных классов-сущностей в специальные системные таблицы `CLASS_DESCRIPTION` (ОПИСАНИЕ КЛАССА) и `OBJECTS_PROP_OBJP` (Свойство класса).

Описание классов-сущностей, поставляемых с *S_Технологией*, уже имеется в указанных таблицах (например “Настройка обмена”, класс-сущность “`TinuTuningExchTeaaEntity`” таблицы “`TUNING_EXCH_TEAA`”), но **все новые классы-сущности нужно загружать дополнительно**. Иначе эти сущности, а следовательно таблицы, будут недоступны таким инструментам *S_Технологии* как «Обмен данными», «Конструктор запросов» и т.д.

Для загрузки описания классов-сущностей предусмотрен специальный пункт меню “Управление системой” – “Настройки разработчика” – “Загрузка описаний классов”. **Внимание!** Пункт меню доступен только пользователю `s_dba`.









Пункт меню открывает форму, содержащую список всех классов-сущностей зарегистрированных в проекте.

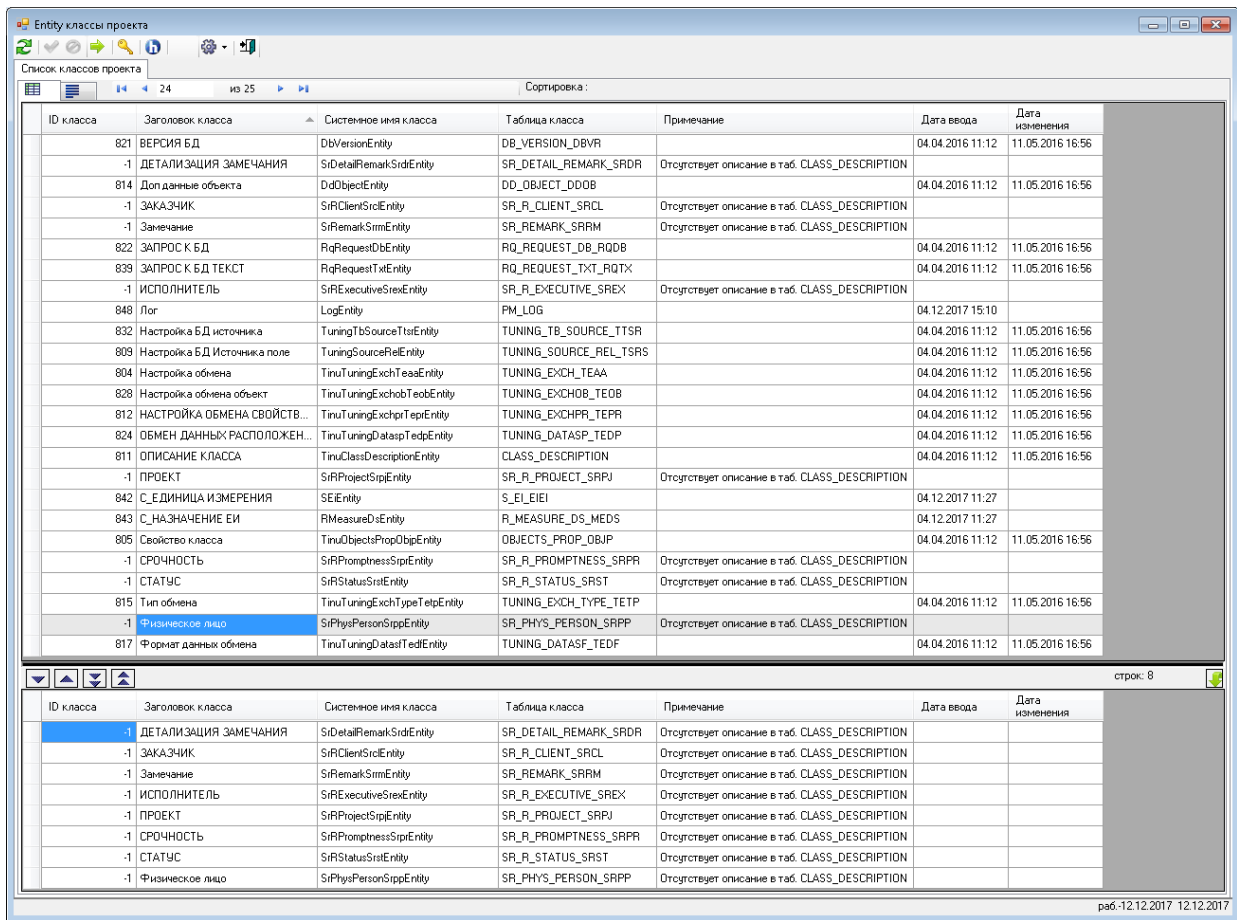
ID класса	Заголовок класса	Системное имя класса	Таблица класса	Примечание	Дата ввода	Дата изменения
821	ВЕРСИЯ БД	DbVersionEntity	DB_VERSION_DBVR		04.04.2016 11:12	11.05.2016 16:56
-1	ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ	SiDetailRemarkSrdtEntity	SR_DETAIL_REMARK_SRDR	Отсутствует описание в таб. CLASS_DESCRIPTION		
814	Доп. данные объекта	DDObjectEntity	DD_OBJECT_DD08		04.04.2016 11:12	11.05.2016 16:56
-1	ЗАКАЗЧИК	SiRClientSrcEntity	SR_R_CLIENT_SRCL	Отсутствует описание в таб. CLASS_DESCRIPTION		
-1	Замечание	SiRemarkSrmEntity	SR_REMARK_SRRM	Отсутствует описание в таб. CLASS_DESCRIPTION		
822	ЗАПРОС К БД	RqRequestDbEntity	RQ_REQUEST_DB_RQDB		04.04.2016 11:12	11.05.2016 16:56
839	ЗАПРОС К БД ТЕКСТ	RqRequestTxtEntity	RQ_REQUEST_TXT_RQTX		04.04.2016 11:12	11.05.2016 16:56
-1	ИСПОЛНИТЕЛЬ	SiRExecutiveSrexEntity	SR_R_EXECUTIVE_SREX	Отсутствует описание в таб. CLASS_DESCRIPTION		
848	Лог	LogEntity	PM_LOG		04.12.2017 15:10	
832	Настройка БД источника	TuningTbSourceTsrEntity	TUNING_TB_SOURCE_TTSR		04.04.2016 11:12	11.05.2016 16:56
809	Настройка БД Источника поле	TuningSourceRelEntity	TUNING_SOURCE_REL_TSRS		04.04.2016 11:12	11.05.2016 16:56
804	Настройка обмена	TinuTuningExchTeaaEntity	TUNING_EXCH_TEEA		04.04.2016 11:12	11.05.2016 16:56
828	Настройка обмена объект	TinuTuningExchobTeobEntity	TUNING_EXCHOB_TEOB		04.04.2016 11:12	11.05.2016 16:56
812	НАСТРОЙКА ОБМЕНА СВОЙСТВ...	TinuTuningExchprTepEntity	TUNING_EXCHPR_TEP		04.04.2016 11:12	11.05.2016 16:56
824	ОБМЕН ДАННЫХ РАСПОЛОЖЕН...	TinuTuningDataspTespEntity	TUNING_DATASP_TEDP		04.04.2016 11:12	11.05.2016 16:56
811	ОПИСАНИЕ КЛАССА	TinuClassDescriptionEntity	CLASS_DESCRIPTION		04.04.2016 11:12	11.05.2016 16:56
-1	ПРОЕКТ	SiRProjectSrpEntity	SR_R_PROJECT_SRPJ	Отсутствует описание в таб. CLASS_DESCRIPTION		
842	С_ЕДИНИЦА ИЗМЕРЕНИЯ	SEEntity	S_EL_EIEI		04.12.2017 11:27	
843	С_НАЗНАЧЕНИЕ ЕИ	RMeasureDsEntity	R_MEASURE_DS_MEDS		04.12.2017 11:27	
805	Свойство класса	TinuObjectsPropObjEntity	OBJECTS_PROP_OBJP		04.04.2016 11:12	11.05.2016 16:56
-1	СПРОЧНОСТЬ	SiRPromptnessSrpEntity	SR_R_PROMPTNESS_SRPR	Отсутствует описание в таб. CLASS_DESCRIPTION		
-1	СТАТУС	SiRStatusSrstEntity	SR_R_STATUS_SRST	Отсутствует описание в таб. CLASS_DESCRIPTION		
815	Тип обмена	TinuTuningExchTypeTetpEntity	TUNING_EXCH_TYPE_TETP		04.04.2016 11:12	11.05.2016 16:56
-1	Физическое лицо	SiPhysPersonSrpEntity	SR_PHYS_PERSON_SRPP	Отсутствует описание в таб. CLASS_DESCRIPTION		
817	Формат данных обмена	TinuTuningDataTfedEntity	TUNING_DATA_SF_TEDF		04.04.2016 11:12	11.05.2016 16:56

На скриншоте видно, что у некоторых сущностей поле «ID класса» имеет значения -1, а поле «Примечание» содержит комментарий – «Отсутствует описание в таб. CLASS_DESCRIPTION». Это и есть сущности, описания которых еще отсутствуют и надо добавить. Если сравнить их *Таблицу класса* с теми таблицами, для которых только что были созданы модули, то окажется, что это и есть те 8 новых классов-сущностей, добавленных в проект. Для того, чтобы загрузить их описания, требуется их выбрать среди остальных классов проекта.

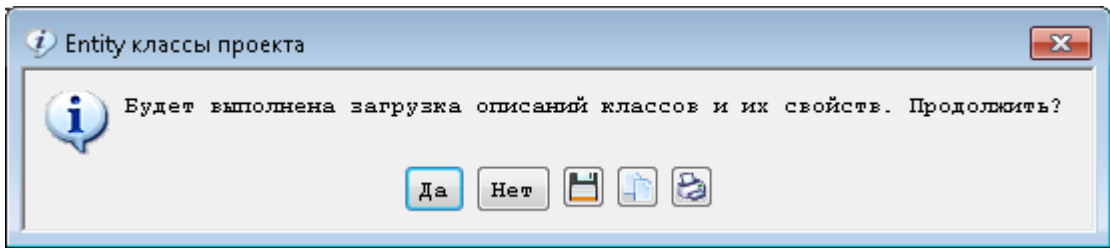
Примечание. Общий принцип выбора нескольких записей в форме-списке.

1. Открыть «Карман» - кнопка  в правом нижнем углу формы.
2. Внизу появляется еще один список, разделенный с основным, панелью с кнопками.
3. На верхнем, основном списке, встаем на нужную запись.
4. Нажать кнопку  на панели между списками. Иногда на верхнем списке при открытом «Кармане», запись может выбираться в «Карман» двойным кликом мыши. Запись «копируется» в нижний список и считается выбранной.
5. Кнопка  «копирует» в «Карман» все записи верхнего списка.
6. Кнопки  и  убирают записи из «Кармана».

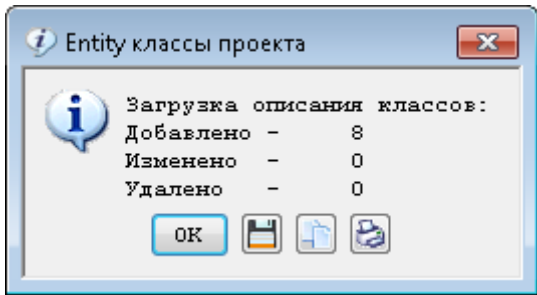
Нужно отобразить в «Карман» классы-сущности с полем «ID класса» = -1. Это могут быть не только новые только что созданные классы-сущности, но и еще какие-то другие (в зависимости от поставки). После того как сущности выбраны, становится доступна кнопка  - **Обновить описание отобранных в карман классов**, нужно ее нажать.



После этого на экране появится диалоговое сообщение с описанием действия и запросом его подтверждения.



После нажатия кнопки «Да» будет выполнен процесс загрузки описаний выбранных классов и их свойств. О ходе выполнения процесса оповещает форма длительного процесса, в конце выдается краткий протокол с результатом.



Далее на форме становятся активны кнопки “Сохранить изменения” и “Отменить изменения”. Чтобы сохранить загруженные описания классов нужно нажать кнопку “Сохранить изменения”. В итоге форма выглядит так.

ID класса	Заголовок класса	Системное имя класса	Таблица класса	Примечание	Дата ввода	Дата изменения
821	ВЕРСИЯ БД	DbVersionEntity	DB_VERSION_DBVR		04.04.2016 11:12	11.05.2016 16:56
849	ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ	SrDetailRemarkSrdtEntity	SR_DETAIL_REMARK_SRDR		12.12.2017 13:01	
814	Доп данные объекта	DdObjectEntity	DD_OBJECT_DDDB		04.04.2016 11:12	11.05.2016 16:56
850	ЗАКАЗЧИК	SrClientSrcEntity	SR_R_CLIENT_SRCL		12.12.2017 13:01	
851	Замечание	SrRemarkSrmEntity	SR_REMARK_SRRM		12.12.2017 13:01	
822	ЗАПРОС К БД	RqRequestDbEntity	RQ_REQUEST_DB_RQDB		04.04.2016 11:12	11.05.2016 16:56
839	ЗАПРОС К БД ТЕКСТ	RqRequestTxEntity	RQ_REQUEST_TXT_RQTX		04.04.2016 11:12	11.05.2016 16:56
852	ИСПОЛНИТЕЛЬ	SrExecutiveSrexEntity	SR_R_EXECUTIVE_SREX		12.12.2017 13:01	
848	Лог	LogEntity	FM_LOG		04.12.2017 15:10	
832	Настройка БД источника	TuningTbSourceTtsrEntity	TUNING_TB_SOURCE_TTSR		04.04.2016 11:12	11.05.2016 16:56
809	Настройка БД Источника поле	TuningSourceRelEntity	TUNING_SOURCE_REL_TSRS		04.04.2016 11:12	11.05.2016 16:56
804	Настройка обмена	TinuTuningExchTteaEntity	TUNING_EXCH_TTEA		04.04.2016 11:12	11.05.2016 16:56
828	Настройка обмена объект	TinuTuningExchobTeobEntity	TUNING_EXCHOB_TEOB		04.04.2016 11:12	11.05.2016 16:56
812	НАСТРОЙКА ОБМЕНА СВОЙСТВ...	TinuTuningExchprTepEntity	TUNING_EXCHPR_TEP		04.04.2016 11:12	11.05.2016 16:56
824	ОБМЕН ДАННЫХ РАСПОЛОЖЕН...	TinuTuningDataspTdpEntity	TUNING_DATASP_TEDP		04.04.2016 11:12	11.05.2016 16:56
811	ОПИСАНИЕ КЛАССА	TinuClassDescriptionEntity	CLASS_DESCRIPTION		04.04.2016 11:12	11.05.2016 16:56
853	ПРОЕКТ	SrProjectSrprEntity	SR_R_PROJECT_SRPR		12.12.2017 13:01	
842	С_ЕДИНИЦА ИЗМЕРЕНИЯ	SEEntity	S_EI_EI		04.12.2017 11:27	
843	С_НАЗНАЧЕНИЕ ЕИ	RMeasureDsEntity	R_MEASURE_DS_MEDS		04.12.2017 11:27	
805	Свойство класса	TinuObjectPropObjpEntity	OBJECTS_PROP_OBJP		04.04.2016 11:12	11.05.2016 16:56
854	СРОЧНОСТЬ	SrRPromptnessSrprEntity	SR_R_PROMPTNESS_SRPR		12.12.2017 13:01	
855	СТАТУС	SrRStatusSrstEntity	SR_R_STATUS_SRST		12.12.2017 13:01	
815	Тип обмена	TinuTuningExchTypeTetpEntity	TUNING_EXCH_TYPE_TETP		04.04.2016 11:12	11.05.2016 16:56
856	Физическое лицо	SrPhysPersonSrppEntity	SR_PHYS_PERSON_SRPP		12.12.2017 13:01	
817	Формат данных обмена	TinuTuningDatastTedfEntity	TUNING_DATASTF_TEDF		04.04.2016 11:12	11.05.2016 16:56

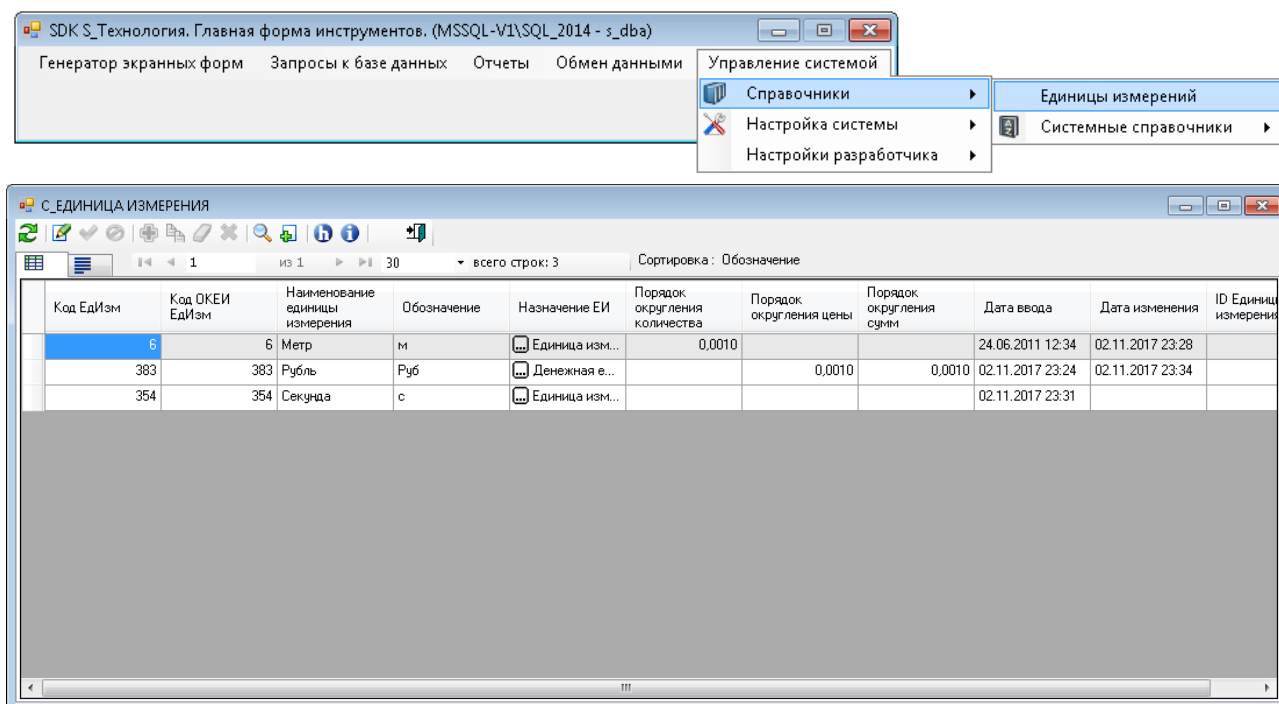
Карман очистился, классам-сущностям, имевших ранее «ID класса» = -1 присвоено реальное значение «ID класса». Описание классов загружено.

Классы-сущности полностью готовы к использованию. Ниже приведен пример простейшего использования класса-сущности.

Шаг 3. Использование таблицы в качестве справочника

В любой достаточно сложной системе имеются специальные таблицы для хранения условно-постоянных данных – Справочники. Справочник допускает стандартную и достаточно упрощенную схему вызова. Обычно достаточно функционала: *Добавить запись, Редактировать запись, Удалить запись*. В S_Технологии все эти действия реализованы на единой для всех Справочников стандартной форме.

В демонстрационном проекте уже имеются несколько Справочников, например, «Единицы измерений»:



Чтобы добавить вызов одной из созданных выше классов-сущностей (например, «Физическое лицо» таблицы SR_PHYS_PERSON_SRPP) как Справочника, можно добавить на главную форму `FrmMain` кнопку `button2` и обработчик `button2_Click`, но рекомендуется использовать для этих целей специальный механизм *S_Технологии* **Меню**. Редактировать меню главной формы проекта нужно Утилитой управления проектом – **pjmgr.exe**, которая входит в состав пакета поставки.

Примечание. Несколько слов об использовании утилиты **pjmgr.exe**.

После инсталляции *S_Технологии*, на рабочем столе будет создана ссылка на группу ярлыков – IM_Tools. В группе имеется ярлык для вызова утилиты **pjmgr.exe** – PJMGR.lnk. К сожалению, пока нет возможности точно настроить все параметры ярлыка, и его придется уточнить вручную.

После инсталляции поле “Объект” ярлыка будет выглядеть примерно так:

"C:\Program Files\IM_Projects\IM_Tools_2.0.0.1_Extended_Демо\PJMGR\pjmgr.exe"

Т.е. будет просто ссылка на exe-файл. Необходимо добавить следующие параметры для запуска этого exe-файла:

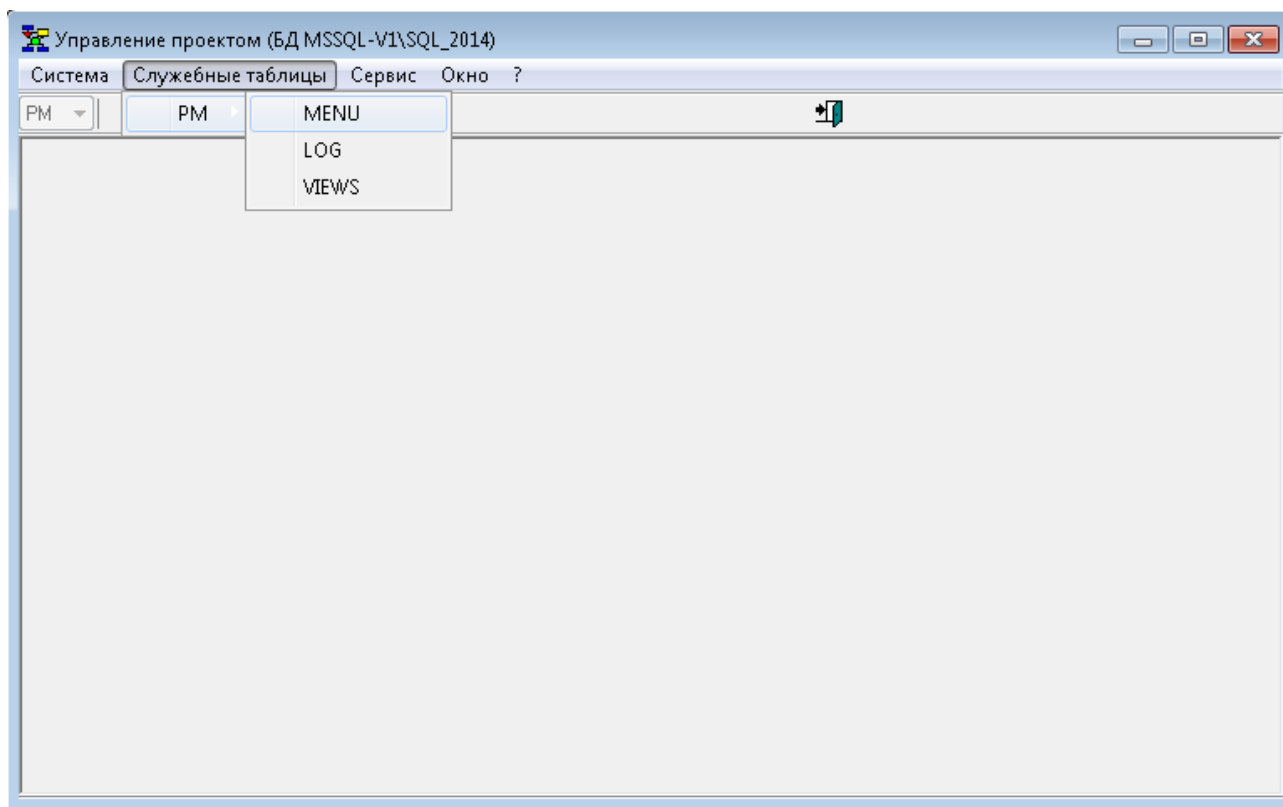
- Для СУБД MSSQL – `-us_dba/<пароль пользователя s_dba> -p<ID VS-проекта (значение поля “Id проекта” = PM)> -a<Имя MSSQL сервера> -tMSSQL -i<Имя экземпляра проекта (значение поля “Id экземпляра проекта”)>`. Таким образом поле “Объект” ярлыка будет выглядеть примерно так:

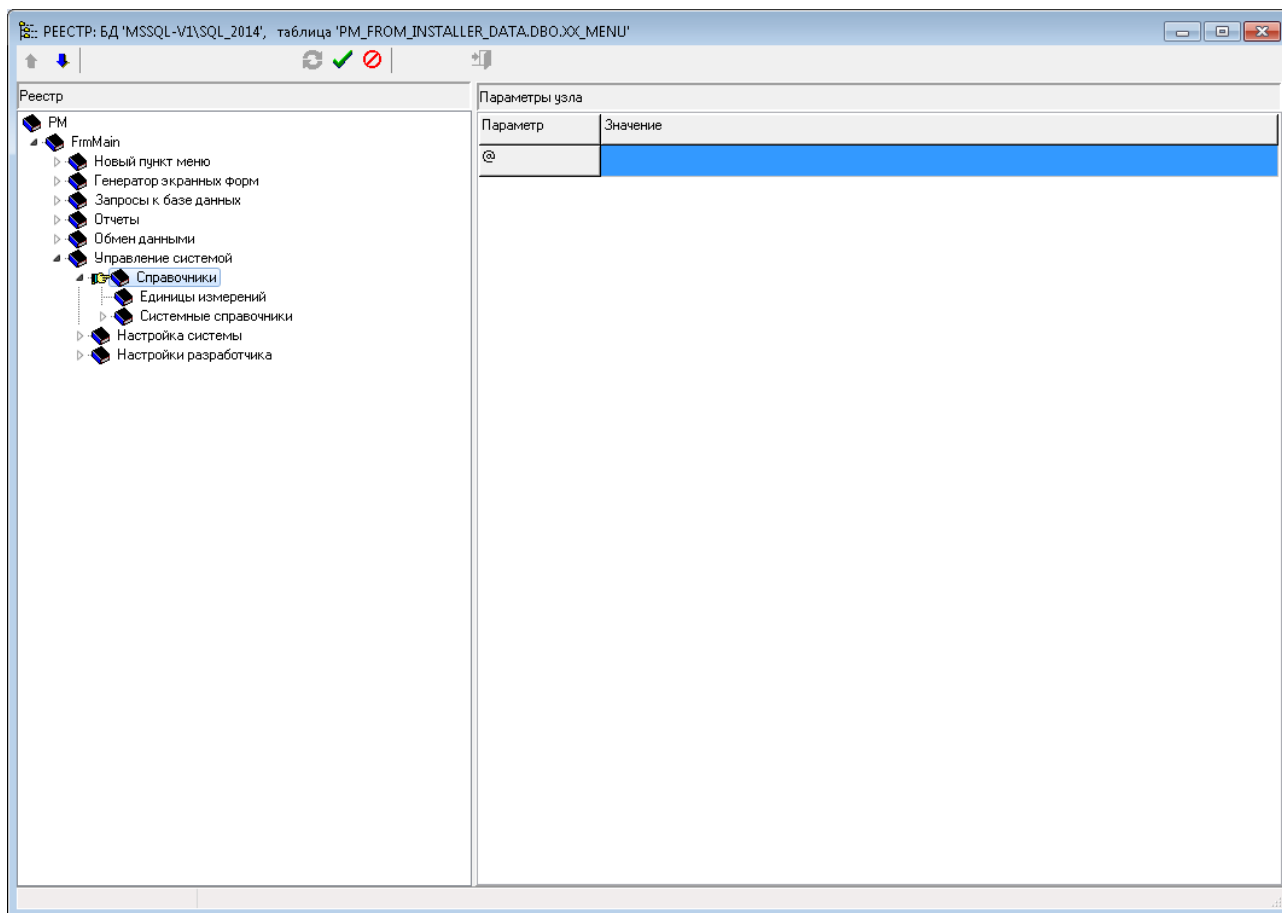
"C:\Program Files\IM_Projects\IM_Tools_2.0.0.1_Extended_Демо\PJMGR\pjmgr.exe" -us_dba/s_admin -pPM -aMSSQL-V1\SQL_2014 -tMSSQL -iPM_FROM_INSTALLER

- Для СУБД Oracle – `-us_dba/<пароль пользователя s_dba> -p<ID VS-проекта (значение поля “Id проекта” = PM)> -a<Имя БД Oracle> -tORACLE`. Таким образом поле “Объект” ярлыка будет выглядеть примерно так:


```
"C:\Program Files\IM_Projects\IM_Tools_2.0.0.1_Extended_Демо\PJMGR\pjmgr.exe" -  
us_dba/s_admin -pPM -aORACLE_DB -tORACLE
```

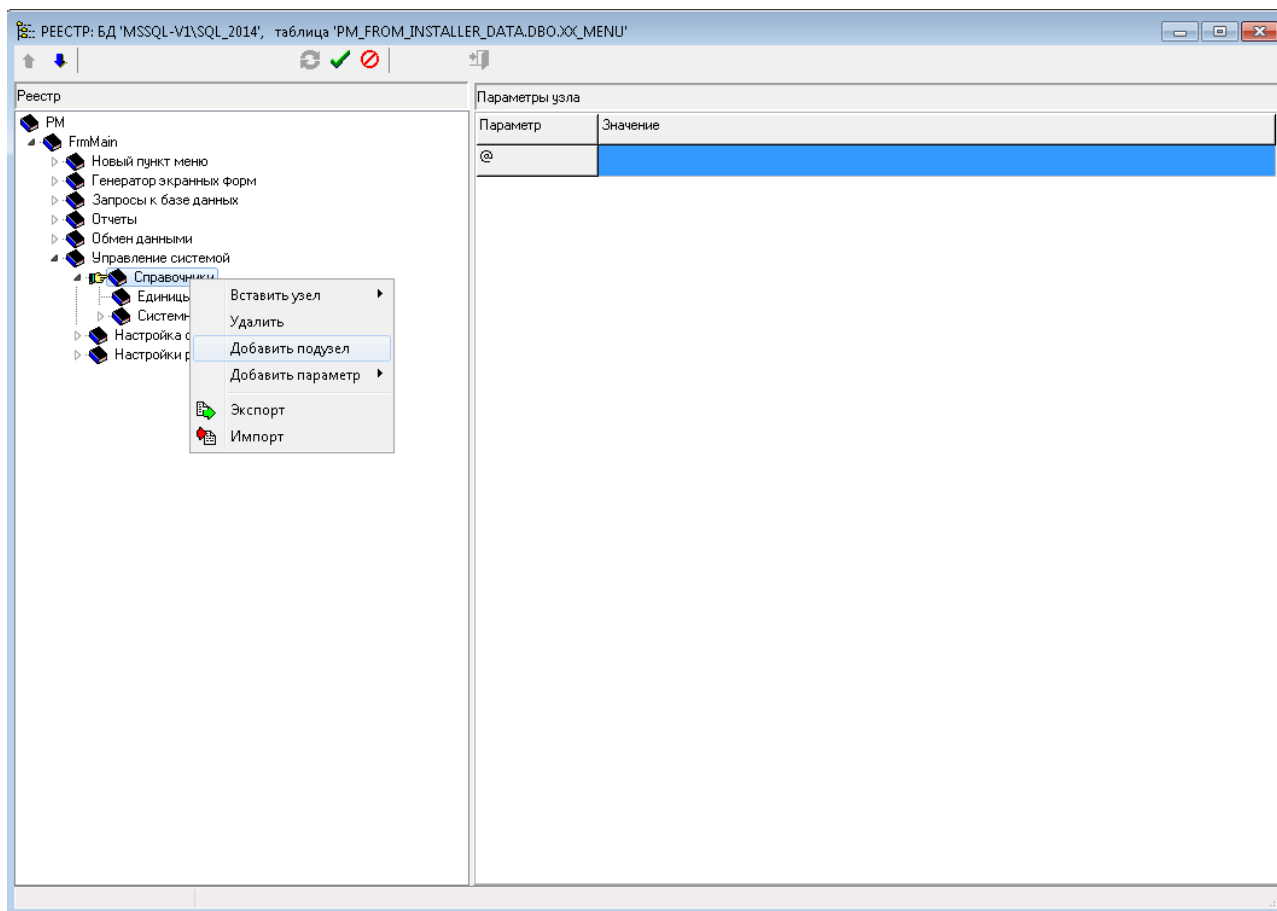
Настроив ярлык утилиты pjmgr.exe, можно переходить к редактированию меню проекта. Для этого в пункте меню “Служебные таблицы” следует выбрать пункт “MENU”:





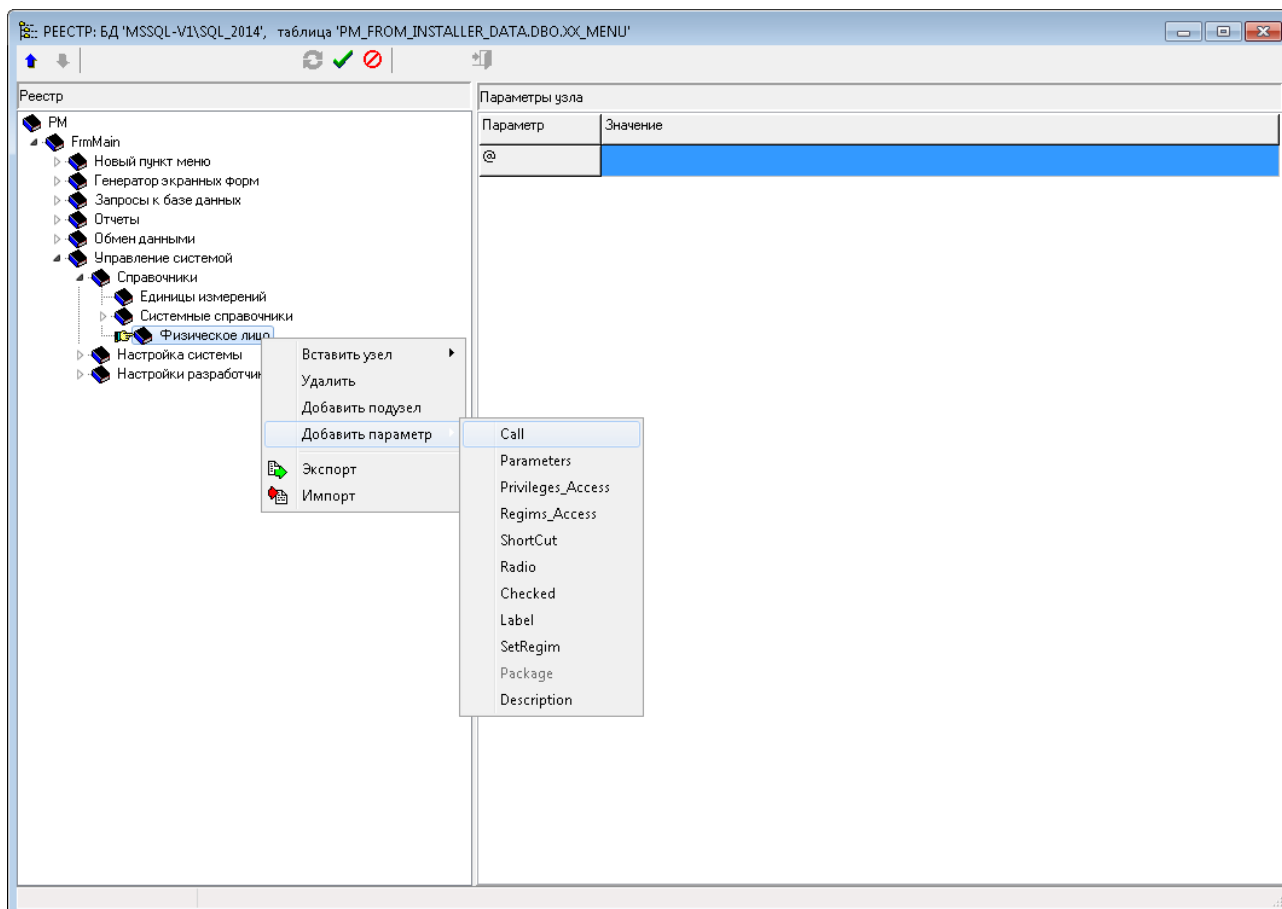
Меню проекта отображается в виде дерева. Узел меню главной формы проекта называется «FrmMain» и совпадает с VS-именем класса главной формы ([FrmMain](#)). Все узлы, кроме «Новый пункт меню» (он не отображается), формируют горизонтальную часть меню, а их подузлы – выпадающие пункты меню.

Нужно нажать кнопку «Начать редактировать» и добавить новый подузел в узел «Справочники». Для этого следует мышью перейти на узел «Справочники», правой клавишей мыши вызвать контекстное меню и выбрать пункт «Добавить подузел».

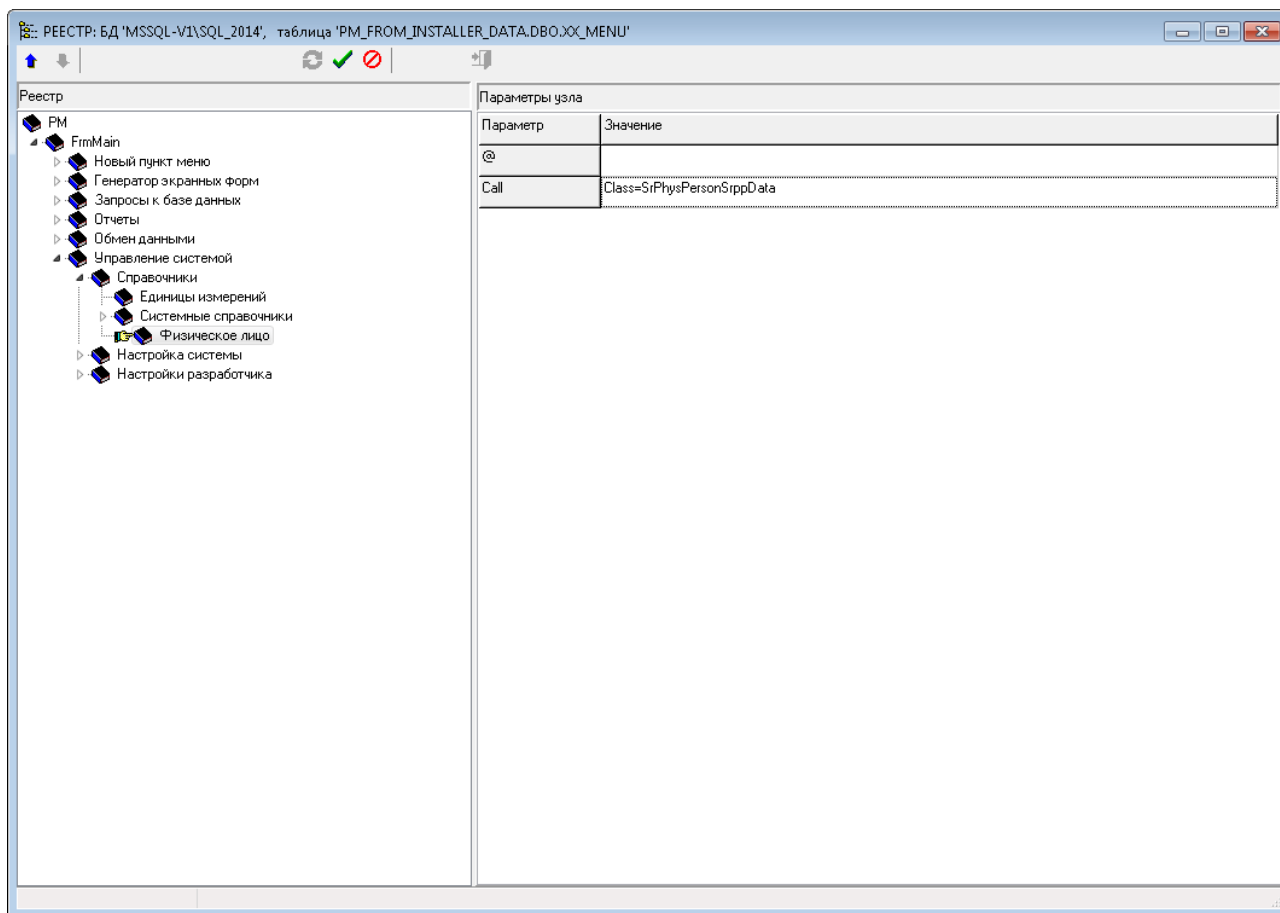


Далее следует ввести его наименование – «Физическое лицо». Этот пункт меню и будет вызывать Справочник физических лиц.

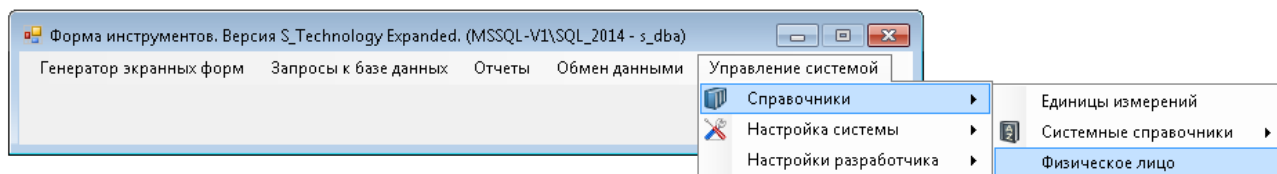
Для того, чтобы пункт меню вызывал класс-сущность как Справочник, ему надо определить имя **Data-класса**, связанного с этой таблицей. Имя Data-класса для таблицы «Физическое лицо» (SR_PHYS_PERSON_SRPP) – [SrPhysPersonSrppData](#). Чтобы определить это имя в меню, следует установить курсор на пункт меню «Физическое лицо», вызвать контекстное меню и выбрать пункт «Добавить параметр»-«Call».

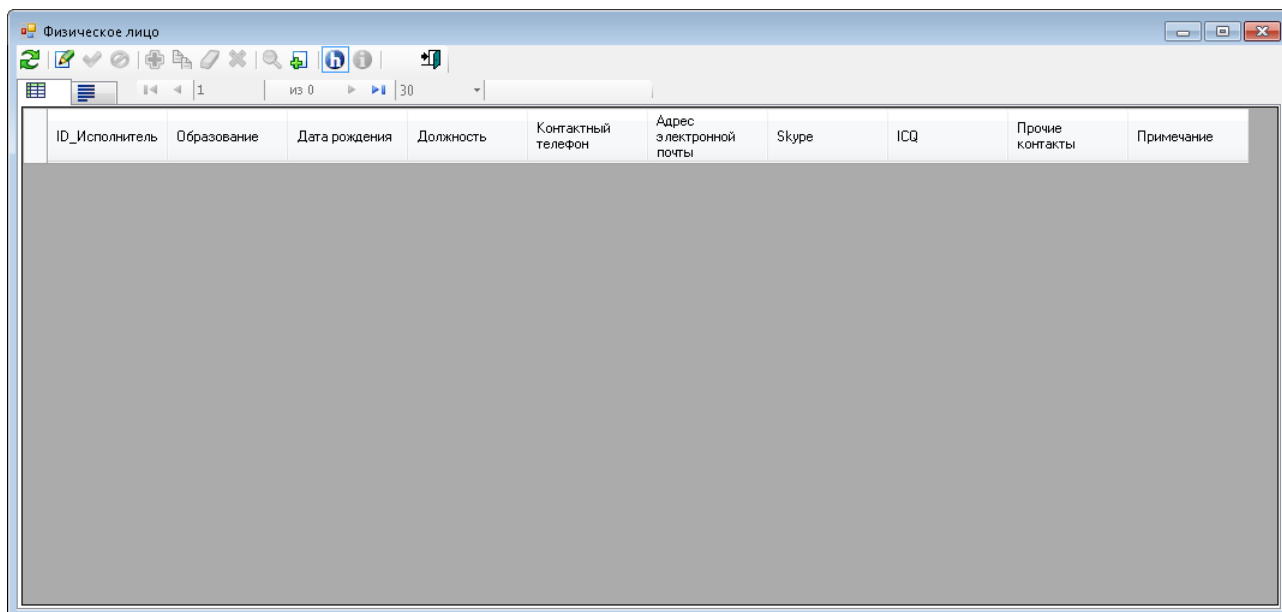


И определить имя Data-класса как ключевую фразу – “Class=<имя Data-класса>” (в текущем примере “Class=SrPhysPersonSrppData”).



Далее нужно подтвердить изменения в меню, сохранив их, после чего можно вернуться к проекту. Если запустить проект и посмотреть меню «Справочники», окажется, что в нем появился новый пункт «Физическое лицо»:





Примечание. Таблица «Физическое лицо» - SR_PHYS_PERSON_SRPP может уже содержать демонстрационные данные (в зависимости от поставки). На скриншоте представлена пустая таблица, но могут быть показаны какие-то записи.

Если для вызова справочника «Физическое лицо» используется кнопка `button2`, в классе главной формы `FrmMain` нужно добавить обработчик `button2_Click`:

```
private void button2_Click(object sender, EventArgs e)
{
    s_View.RefExec("SrPhysPersonSrppData", null, null, 0, false, string.Empty, null);
}
```

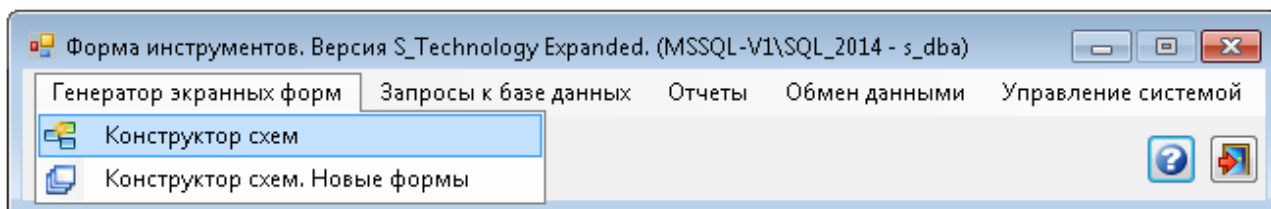
Примечание. Более подробно о методе `s_View.RefExec` см. в справке `S_.chm`.

Через форму справочника можно выполнять добавление, корректировку или удаление записей в таблице «Физическое лицо» - SR_PHYS_PERSON_SRPP.

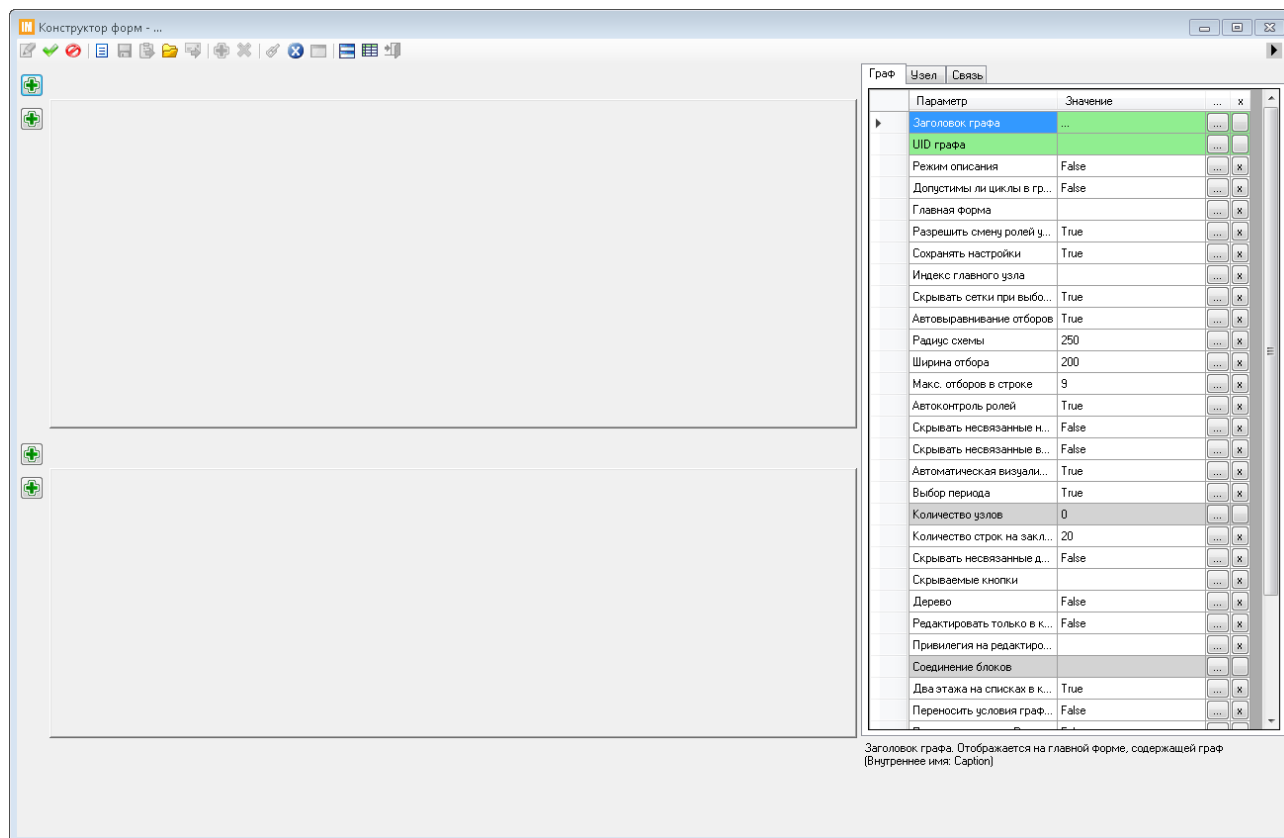
Это пример самого простого использования модулей таблиц (сущностей). Далее будет показано создание более сложных форм, визуализирующих таблицы.

Шаг 4. Создание формы при помощи Конструктора форм

Для запуска Конструктора форм нужно вызвать приложение и выбрать пункт меню «Генератор экранных форм» -> «Конструктор схем».



Откроется специальная форма, предназначенная для конструирования новых форм проекта.



Форма открыта сразу в режиме редактирования и содержит следующие элементы.

1. Кнопки управления.

- «Начать редактирование» - переход в режим редактирования графа.
- «Сохранить изменения» - сохранить граф в памяти, чтобы иметь возможность откатиться к сохраненному состоянию, если это потребуется.
- «Отменить изменения» - откатиться к последнему сохраненному состоянию графа.
- «Создать новую XML-схему» - создать новый граф.
- «Сохранить XML-схему в файле» - сохранить XML-схему графа в файле.
- «Добавить схему к проекту» - добавить XML-схему графа к одному из проектов решения. (К проекту будет добавлена сама схема, а также автоматически созданные формы для каждого узла графа, у которых указана сущность).
- «Загрузить XML-схему» - загрузить граф из XML-схемы графа, хранящейся в файле *.xml.
- «Добавить новый узел» - добавить в граф новый узел. Кнопки «+» слева от каждой панели в режиме «Форма/схема» (по умолчанию). Или кнопка вверх в режиме «Блок/схема».
- «Удалить выбранный узел» - удалить из графа выбранный узел.
- «Создать связи из БД» - автоматически добавить в граф все связи, какие он сможет найти в модулях таблиц (сущностях) уже добавленных в граф узлов.
- «Свойства: все/важные» - изменяет режим отображения свойств.
- «Показать форму на основе XML-схемы» - осуществить предварительный просмотр формы, которую мы строим конструктором форм.
- «Режим блок/схема» - переход в режим построителя запросов к БД.
- «Режим форма/схема» - показ создаваемой формы в режиме макета формы или схемы графа. **Текущий режим «Форма».**

2. Панели для размещения «Узлов» (по центру), описание сверху вниз.

- Панель для размещения «Узлов» типа «Фильтр/Отбор» главной формы.
- Панель для «Узлов» главных форм.
- Панель для размещения «Узлов» типа «Фильтр/Отбор» дочерней формы.
- Панель для «Узлов» дочерних форм.

3. Таблконтрол с таблицами свойств (справа) с тремя закладками.

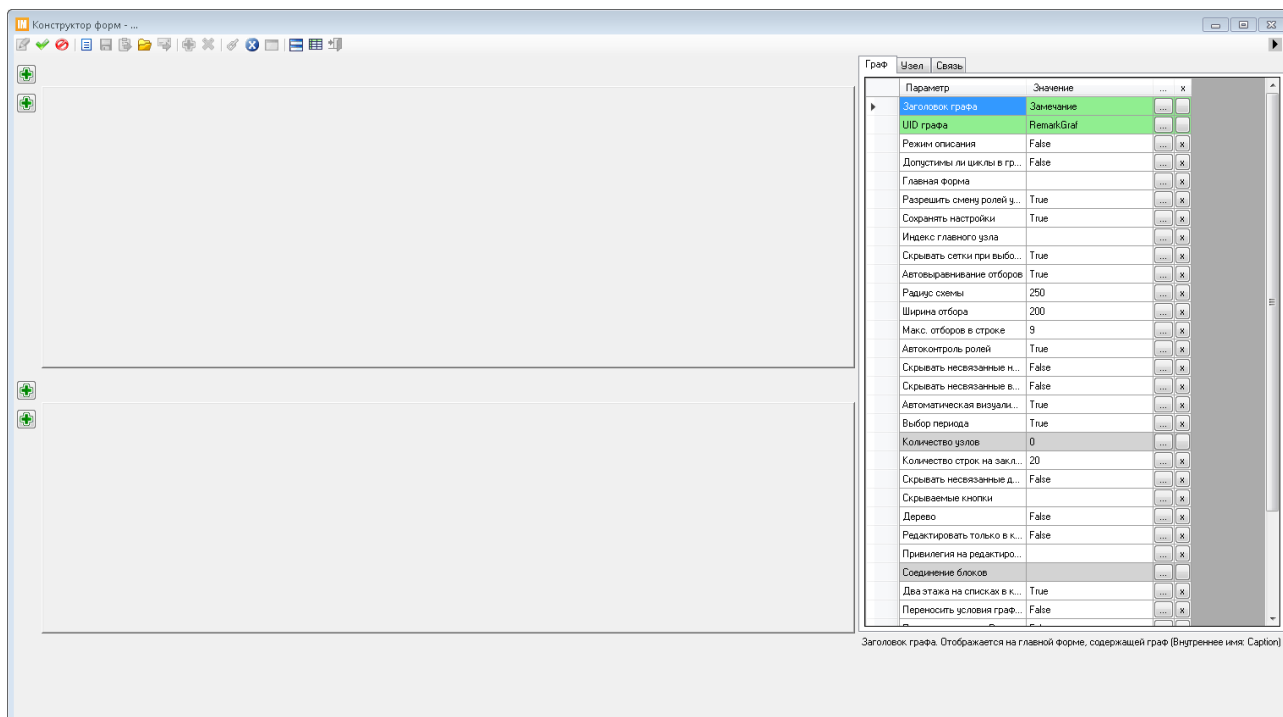
- «Граф» – таблица со свойствами графа.
- «Узел» – таблица со свойствами активного узла «Графа», выбранного пользователем.
- «Связь» – таблица со свойствами связи, выбранной пользователем.
 - а) Кнопка «Соединить два узла» - создает связь между узлами.
 - б) Кнопка «Разорвать связь между узлами» - удаляет выбранную пользователем связь между узлами.

Немного терминологии.

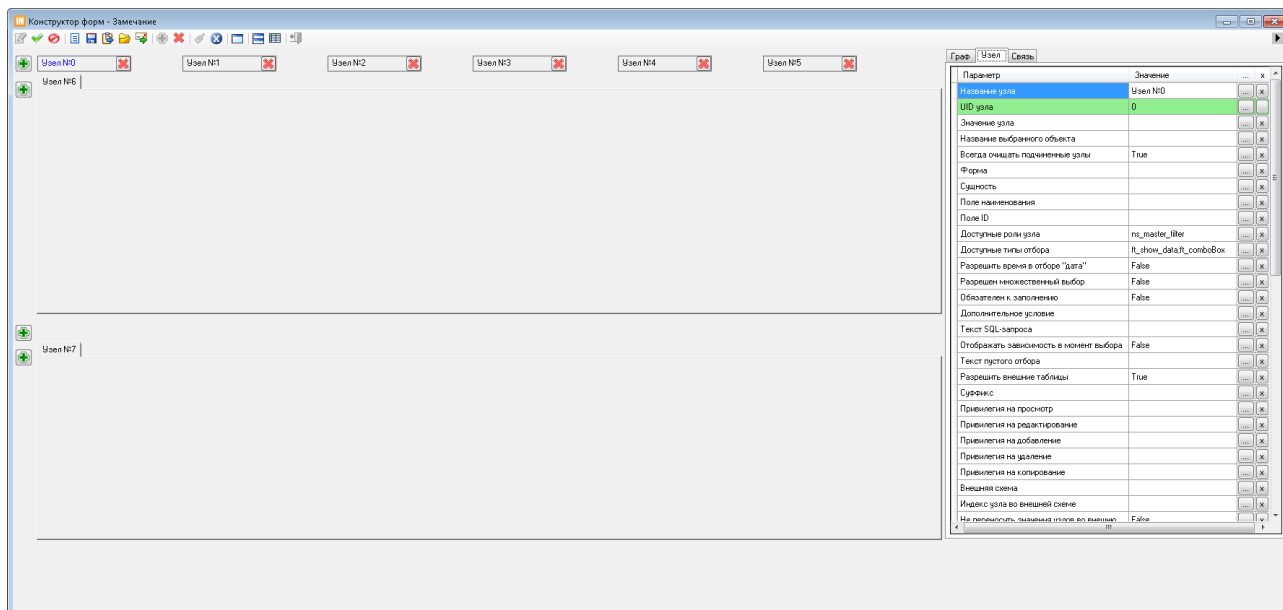
- **«Граф»**. Объект, содержащий конструируемые элементы – «Узлы» и «Связи» между ними. «Граф» может содержать одну форму-узел или множество. В форме «Конструктор форм» выполняется создание/редактирование описания «Графа». Описание «Графа» реализуется в виде VS-модуля, содержащего XML-схему («Схема графа» или просто «Схема»). На основании XML-схемы описания «Графа» и строится объект граф со всеми описанными формами, сущностями, связями.
- **«Узел»**. Составляющая часть «Графа». Реализация формы или фильтра на базе конкретной таблицы (Сущности).
- **«Связь»**. Составляющая часть «Графа». Реализация связей между «Узлами» (таблицами) «Графа».
- **«Фильтр»** или **«Отбор»**. Особый тип реализации «Узла». Отображается в виде компактной панели с кнопкой. Позволяет выбрать одно или несколько значений из таблицы и по ним отфильтровать связанный с ним дочерний узел.

В качестве примера можно построить граф «Замечание» на основе демонстрационных таблиц и созданных на Шаге 3 сущностей к ним. Граф будет состоять из мастер-формы сущности «ЗАМЕЧАНИЕ» (таблица SR_REMARK_SRRM), детальной формы сущности «ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ» (таблица SR_DETAIL_REMARK_SRDR) и шести фильтров (по-другому отборов). Четыре фильтра будут реализованы на базе сущностей: «ПРОЕКТ» (таблица SR_R_PROJECT_SRPJ), «СТАТУС» (таблица SR_R_STATUS_SRST), «СРОЧНОСТЬ» (таблица SR_R_PROMPTNESS_SRPR) и «ИСПОЛНИТЕЛЬ» (таблица SR_R_EXECUTIVE_SREX). Еще будут два специальных фильтра по «Дата создания» записи сущности «ЗАМЕЧАНИЕ» (поле SR_REMARK_SRRM.DATE_CREATE_SRRM): «Дата от» и «Дата по».

Первым делом, на закладке свойств графа следует заполнить параметры *Заголовок графа* - «Замечание» и *UID графа* - «RemarkGraf» (значения параметров произвольные). Другие свойства графа пока изменять не нужно.



Для создания 6-ти узлов-фильтров для главной формы нужно нажать самую верхнюю кнопку (+) шесть раз. Нажатие второй сверху кнопки (+) создаст узел главной формы. А нижняя кнопка (+) создаст узел детальной формы.



Далее нужно выбрать сущности для каждого созданного узла.

Для того чтобы выбрать сущность для первого «Узла №0» следует выбрать этот узел щелчком мыши, перейти на закладку «Узел» и нажать на кнопку с многоточием (...) напротив свойства узла «Сущность». Появится форма со списком всех сущностей, зарегистрированных в системе. Среди них нужно найти строку с названием сущности «ПРОЕКТ» и выйти из формы с выбором, нажав на кнопку «Выход с выбором текущей записи».

Выбор свойства "Сущность"

13 из 25

№ п/п	Наименование	Описание	Дата-класс	Таблица БД
0	ВЕРСИЯ БД	DbVersionEntity	DbVersionData	DB_VERSION_DBVR
1	ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ	SrDetailRemarkSrdrEntity	SrDetailRemarkSrdrData	SR_DETAIL_REMARK_SRDR
2	Доп данные объекта	DdObjectEntity	DdObjectData	DD_OBJECT_DDDB
3	ЗАКАЗЧИК	SrRClientSrcEntity	SrRClientSrcData	SR_R_CLIENT_SRCL
4	ЗАПРОС К БД ТЕКСТ	RqRequestTxtEntity	RqRequestTxtData	RQ_REQUEST_TXT_RQTX
5	ЗАПРОС К БД	RqRequestDbEntity	RqRequestDbData	RQ_REQUEST_DB_RQDB
6	ИСПОЛНИТЕЛЬ	SrRExecutiveSrexEntity	SrRExecutiveSrexData	SR_R_EXECUTIVE_SREX
7	Лог	LogEntity	LogData	PM_LOG
8	Настройка обмена объект	TinuTuningExchobTeobEntity	TInuTuningExchobTeobData	TUNING_EXCHOB_TEOB
9	НАСТРОЙКА ОБМЕНА СВОЙСТ...	TinuTuningExchprTepEntity	TInuTuningExchprTepData	TUNING_EXCHPR_TEP
10	Настройка обмена	TinuTuningExchTeaaEntity	TInuTuningExchTeaaData	TUNING_EXCH_TEAA
11	ОПИСАНИЕ КЛАССА	TinuClassDescriptionEntity	TInuClassDescriptionData	CLASS_DESCRIPTION
12	ПРОЕКТ	SrRProjectSrpiEntity	SrRProjectSrpiData	SR_R_PROJECT_SRPI
13	Свойство класса	TinuObjectsPropObjEntity	TInuObjectsPropObjData	OBJECTS_PROP_OBJP
14	СЗ_С_СРОЧНОСТЬ	SrRPromptnessSrprEntity	SrRPromptnessSrprData	SR_R_PROMPTNESS_SRPR
15	СТАТУС	SrRStatusSrstEntity	SrRStatusSrstData	SR_R_STATUS_SRST
16	Физическое лицо	SrPhysPersonSrppEntity	SrPhysPersonSrppData	SR_PHYS_PERSON_SRPP
17	SR_REMARK_SRRM	SrRemarkSrmEntity	SrRemarkSrmData	SR_REMARK_SRRM
18	Формат данных обмена	TinuTuningDataSFtedfEntity	TInuTuningDataSFtedfData	TUNING_DATA_SF_TEDF
19	С_ЕДИНИЦА ИЗМЕРЕНИЯ	SEiEntity	SEiData	S_EI_EIEI
20	ОБМЕН ДАННЫХ РАСПОЛОЖЕ...	TinuTuningDataSPtedpEntity	TInuTuningDataSPtedpData	TUNING_DATA_SP_TEDP
21	Настройка БД Источника поле	TuningSourceRelEntity	TuningSourceRelData	TUNING_SOURCE_REL_TSRS
22	Тип обмена	TinuTuningExchTypeTetpEntity	TInuTuningExchTypeTetp...	TUNING_EXCH_TYPE_TETP
23	Настройка БД источника	TuningTbSourceTtsrEntity	TuningTbSourceTtsrData	TUNING_TB_SOURCE_TTSR
24	С_НАЗНАЧЕНИЕ ЕИ	RMeasureDsEntity	RMeasureDsData	R_MEASURE_DS_MEDS

Некоторые свойства узла автоматически заполняются.

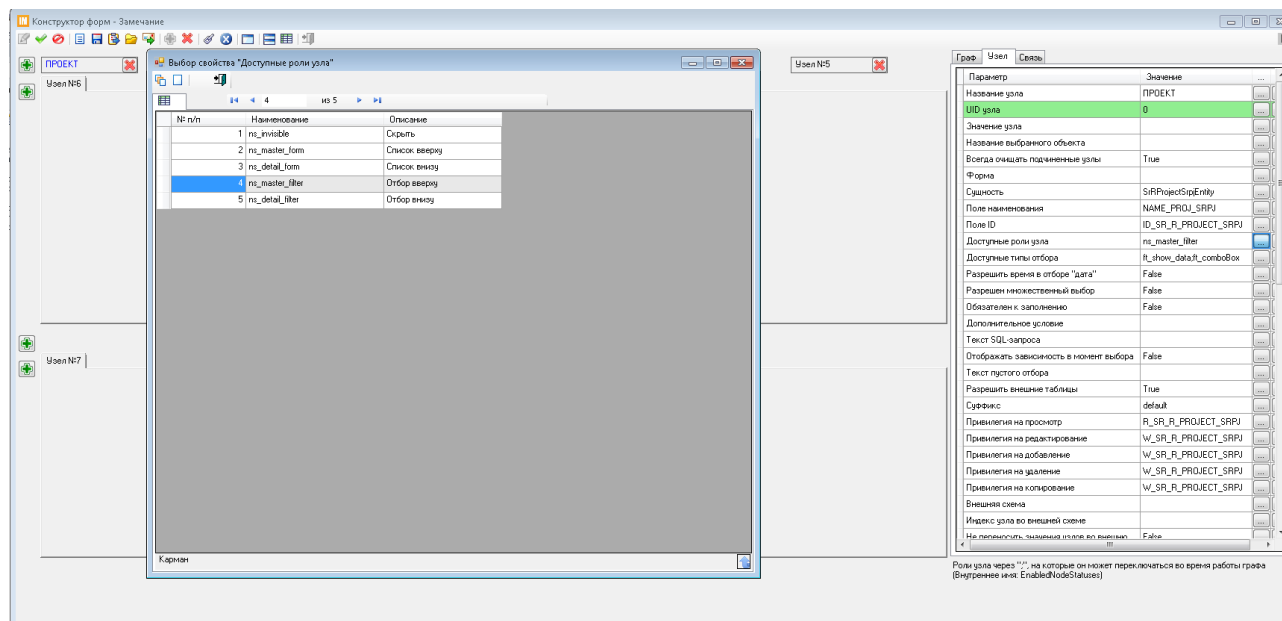
Конструктор форм - Замечание

ПРОЕКТ Узел N1 Узел N2 Узел N3 Узел N4 Узел N5 Узел N6 Узел N7

Граф Узел Свой

Параметр	Значение
Название узла	ПРОЕКТ
UID узла	0
Значение узла	
Название выбранного объекта	
Всегда очищать подчиненные узлы	True
Форма	
Сущность	SrRProjectSrpiEntity
Поле наименования	NAME_PROJ_SRPI
Поле ID	ID_SR_R_PROJECT_SRPI
Доступные роли узла	rs_master_filter
Доступные типы отбора	It_show_data_filter_combobox
Разрешить время в отборе "дата"	False
Разрешить множественный выбор	False
Обязателен к заполнению	False
Дополнительное условие	
Текст SQL-запроса	
Обрабатывать зависимость в момент выбора	False
Текст пустого отбора	
Разрешить вставку таблицы	True
Срефлекс	default
Привязка на просмотр	R_SR_R_PROJECT_SRPI
Привязка на редактирование	W_SR_R_PROJECT_SRPI
Привязка на добавление	W_SR_R_PROJECT_SRPI
Привязка на удаление	W_SR_R_PROJECT_SRPI
Привязка на копирование	W_SR_R_PROJECT_SRPI
Внешняя смена	
Индикатор узла во внешней смене	
На переключатель, позволяющий переключаться	False

Возможно, иногда потребуется изменить свойство «Доступные роли узла». Для этого необходимо в открывшейся форме со списком доступных ролей узла выбрать одну роль «Отбор вверху» (если в открывшейся форме был распахнут карман, закрыть его).



В данном примере это не требуется, т.к. узел размещен на панели для размещения «Узлов» типа «Фильтр» главной формы и ему автоматически присвоена доступная роль «Отбор вверху» (ns_master_filter). Остальные свойства пока менять не нужно.

Аналогичным образом нужно назначить сущности и остальным узлам.

Чтобы назначить роль второго узла нужно выбрать узел «Узел №1» и указать для него сущность «СТАТУС», «Доступные роли узла» = «Отбор вверху» (ns_master_filter).

Для назначения роли третьего узла следует выбрать «Узел №2» и указать для него сущность «СРОЧНОСТЬ», «Доступные роли узла» = «Отбор вверху» (ns_master_filter).

Четвертому узлу «Узел №3» назначим сущность «ИСПОЛНИТЕЛЬ», «Доступные роли узла» = «Отбор вверху» (ns_master_filter).

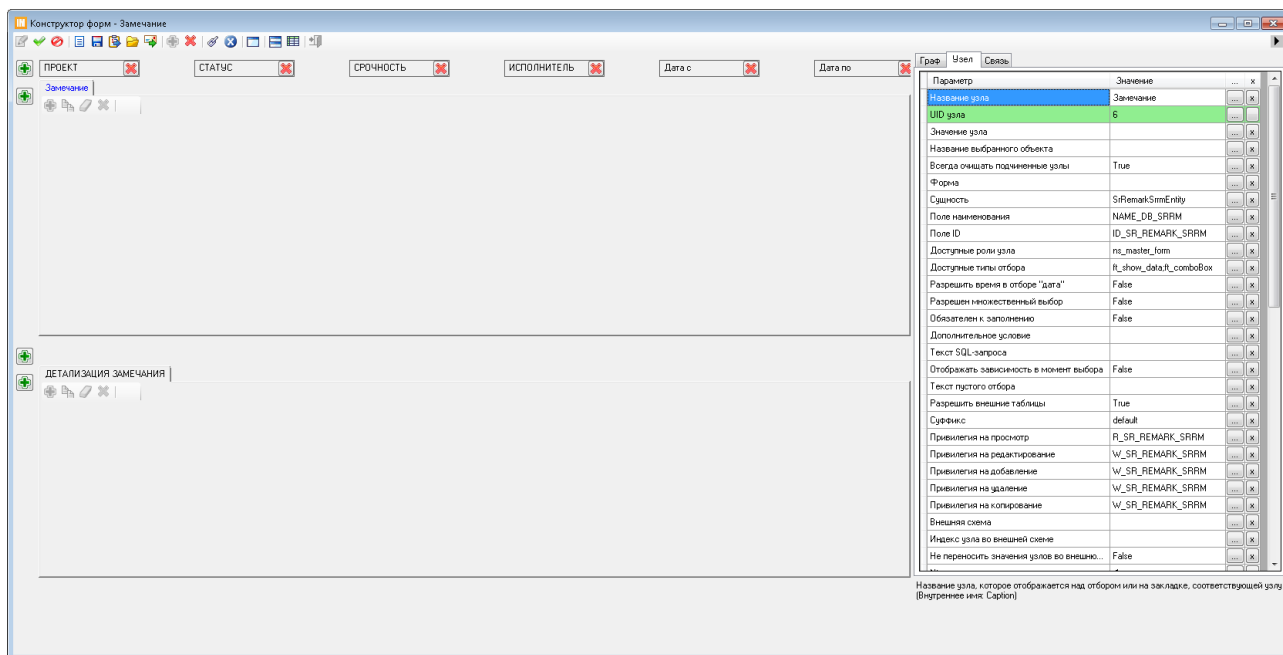
Для пятого и шестого узлов сущность выбирать не нужно. Вместо этого следует выбрать узел «Узел №4», «Доступные роли узла» = «Отбор вверху» (ns_master_filter). И установить свойство «Доступные типы отбора» в значение «Дата» (ft_date). А также изменить свойство «Название узла», задав значение «Дата с». Для узла № «Узел №5» все параметры настраиваются так же, только «Название узла» будет иное – «Дата по».

Теперь нужно назначить сущности узлам форм.

Для назначения сущности главной форме следует выбрать узел «Узел №6» и назначить ему сущность «Замечание», «Доступные роли узла» = «Список вверху» (ns_master_form) – главная форма (это значение будет присвоено автоматически).

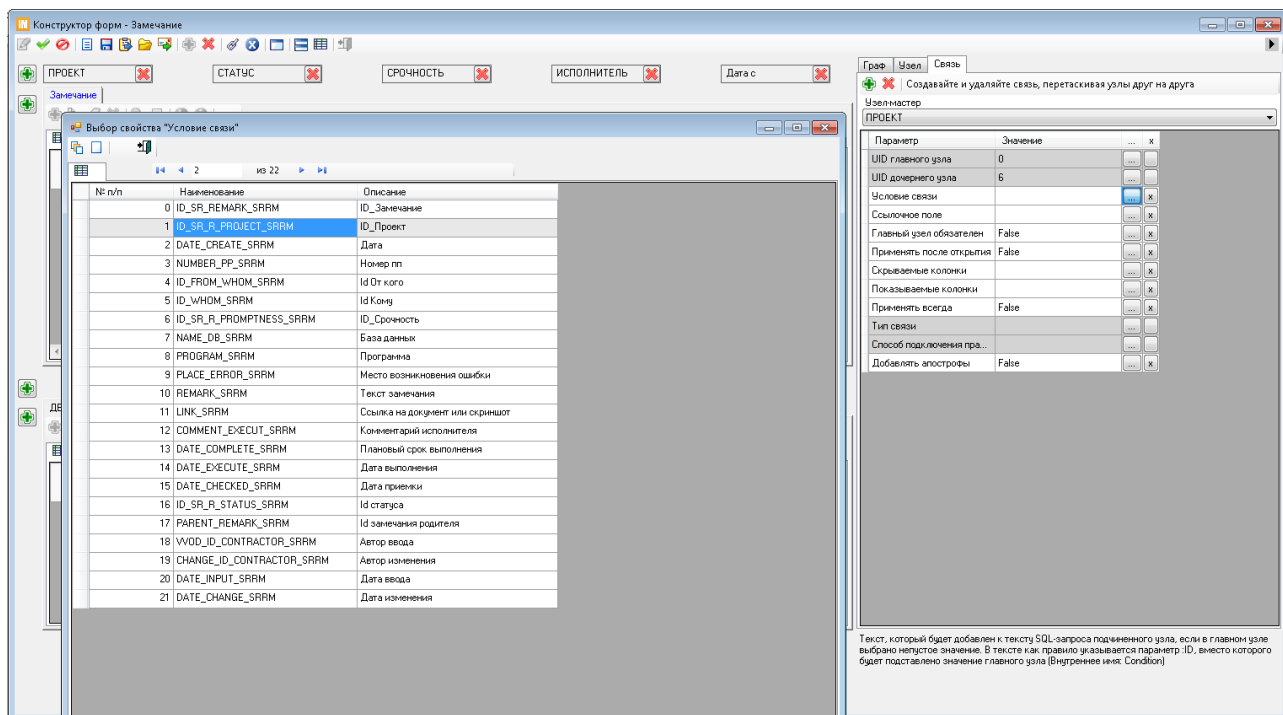
Для назначения сущности дочерней форме следует выбрать узел «Узел №7» и назначить ему сущность «ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ», «Доступные роли узла» = «Список внизу» (ns_detail_form) – дочерняя форма (это значение будет присвоено автоматически).

В результате граф будет выглядеть следующим образом.



После того как все узлы графа определены, требуется связать узлы между собой. Для этого нужно выбрать узел, для которого будут определяться узлы-родители, и перейти на закладку «Связь».

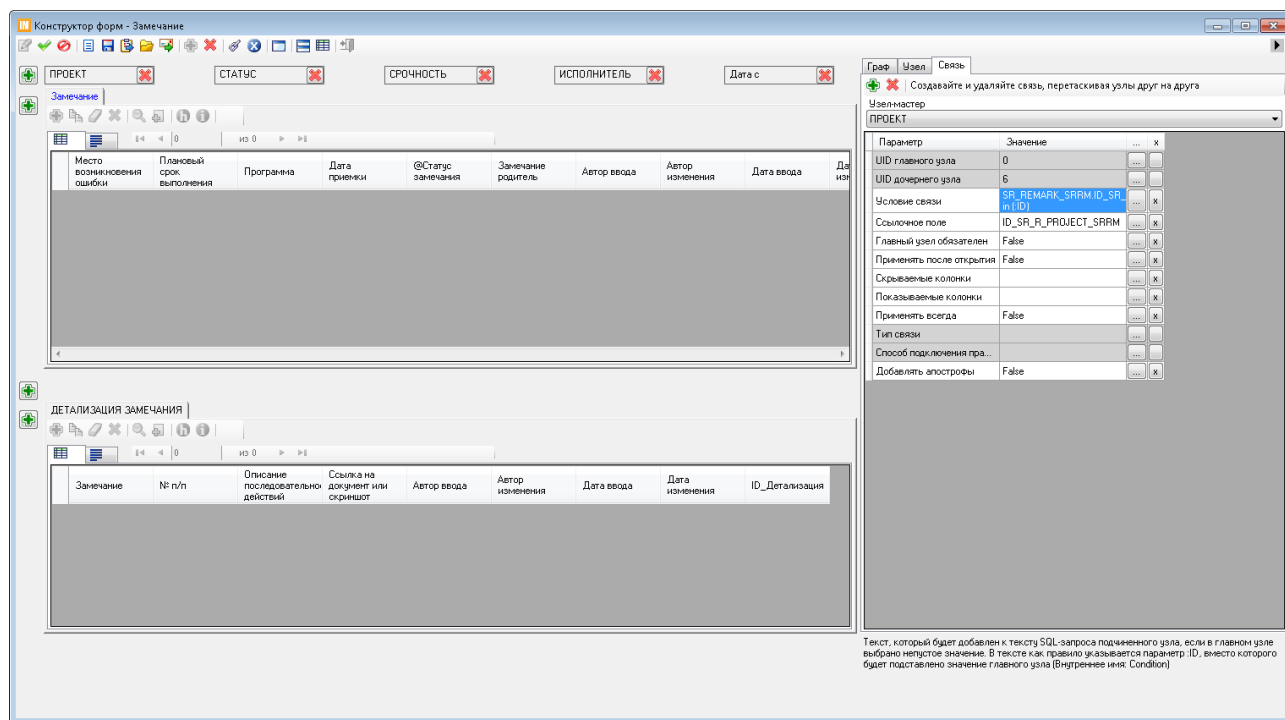
Начать можно, например, с узла «Замечание», выбрав его кликом мыши. Из выпадающего списка на вкладке «Связь» выпадит узел-мастер «ПРОЕКТ» и нажать кнопку «Соединить два узла». Появятся свойства созданной связи. Необходимо установить значение свойству «Условие связи», щелкнув по кнопке (...).



В открывшейся форме перечислены атрибуты сущности «Замечание», среди них нужно выбрать атрибут «ID_SR_R_PROJECT_SRRM» (ID_Проект). Именно это поле в БД является

FOREIGN KEY, связывающим таблицы «ЗАМЕЧАНИЕ» и «ПРОЕКТ». Хотя технически связующее поле не обязательно должно быть FOREIGN KEY.

В результате в поле «Условие связи» должна появиться строка «SR_REMARK_SRRM.ID_SR_R_PROJECT_SRRM in (:ID)». При желании, пользователь может вручную писать SQL-условие связи между узлами. Вместо «:ID» программа будет подставлять значение узла-мастера, в данном случае ID текущей записи (записей) узла «ПРОЕКТ».



Аналогично строятся связи для остальных узлов.

Для связи «Замечание» - «СТАТУС» для свойства «Условие связи» нужно выбрать атрибут «ID_SR_R_STATUS_SRRM» (Id статуса).

Для связи «Замечание» - «СРОЧНОСТЬ» для свойства «Условие связи» нужно выбрать атрибут «ID_SR_R_PROMPTNESS_SRRM» (ID_Срочность).

Для связи «Замечание» - «ИСПОЛНИТЕЛЬ» для свойства «Условие связи» нужно выбрать атрибут «ID_WHOM_SRRM» (Id Кому).

Для связи «Замечание» - «Дата с» для свойства «Условие связи» нужно выбрать атрибут «DATE_CREATE_SRRM» (Дата создания). А затем вручную скорректировать «Условие связи», установив его в «SR_REMARK_SRRM.DATE_CREATE_SRRM >= :ID».

Для связи «Замечание» - «Дата по» для свойства «Условие связи» так же нужно выбрать атрибут «DATE_CREATE_SRRM» (Дата создания). А затем вручную скорректировать «Условие связи», установив его в «SR_REMARK_SRRM.DATE_CREATE_SRRM <= :ID».

Последняя связь между главной и дочерней формам «ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ» - «Замечание». Для этой связи для свойства «Условие связи» нужно выбрать атрибут «ID_SR_REMARK_SRDR» (ID_Замечание).

После того как все узлы и их связи определены описание графа считается готовым, нужно нажать кнопку «Сохранить изменения» и попробовать вызвать форму, созданную на основе графа, используя кнопку «Показать форму на основе XML-схемы». Должна получиться следующая форма (см. скриншот ниже).

Если предварительный просмотр формы показал что что-то сделано неверно, то можно вернуться в конструктор форм и сразу же исправить ошибку.

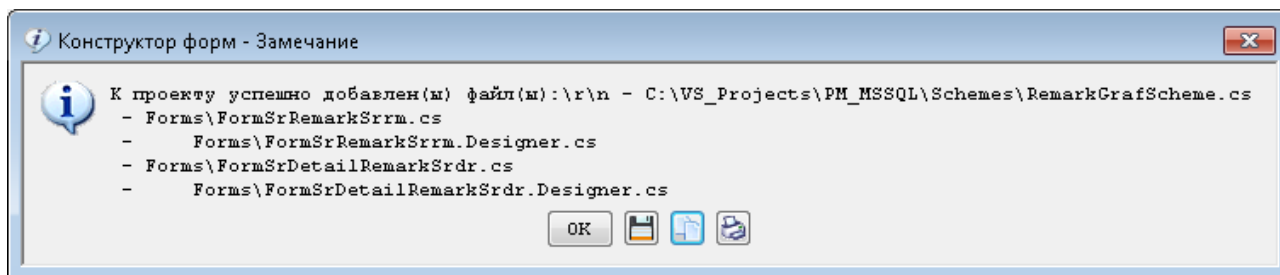
Примечание. Демонстрационные таблицы могут содержать тестовые данные (в зависимости от поставки). При открытии формы, данные в ней показываюся не сразу. У пользователя есть возможность настроить «Отборы» и потом нажать кнопку «Обновить». Такое поведение форм предназначено для открытия больших выборок, и регулируется свойством графа – «Скрывать сетки при выборе». По умолчанию это свойство имеет значение true. Если требуется немедленный показ данных (и есть уверенность что выборки будут небольшие), можно установить свойство графа «Скрывать сетки при выборе» в false (см. Шаг 6, редактирование графа). На скриншоте представлена пустая таблица, но могут быть показаны какие-то записи.

Шаг 5. Подключение формы-графа к проекту

Возможны два сценария работы с созданным графом.

Сценарий 1. Автоматическое добавление графа к проекту

Чтобы автоматически добавить граф к проекту нужно на форме конструктора форм нажать кнопку «Добавить схему к проекту» и выбрать текущий проект (в данном примере проект PM_MSSQL.csproj). Появится информационное сообщение о добавлении формы к проекту.

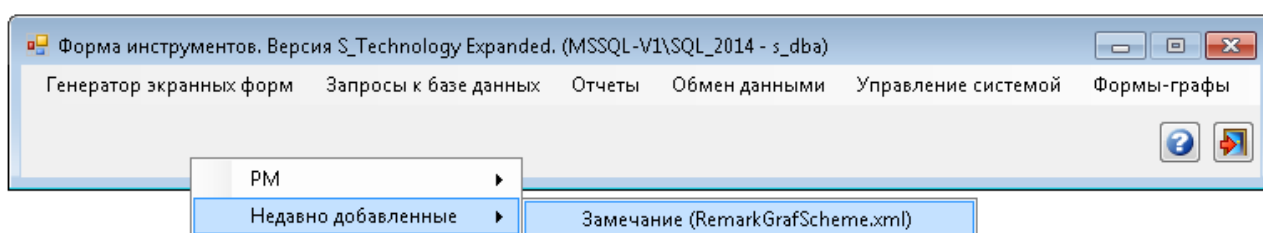


Модуль RemarkGrafScheme.cs содержит класс RemarkGraf, в котором находится описание (xml-схема) графа 'Замечание'. В дальнейшем, при вызове модуля на основе этого описания будет построен сам граф, со всеми необходимыми классами-сущностями, формами и остальными элементами.

Модули FormSrRemarkSrrm.cs и FormSrDetailRemarkSrdr.cs реализуют формы для узлов-списков «Замечание» и «ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ». Первоначально классы форм практически пустые. Они созданы для дальнейшего расширения функциональности узла, который отображается формой.

После подключения графа проект надо пересобрать.

Имеется специальный пункт меню для оперативного вызова форм: «Генератор экранных форм» – «Конструктор схем. Новые формы». При его выборе формируется и показывается контекстное меню из форм, недавно добавленных в проект.



Формы-графы, недавно добавленные в проект, хранятся в папке запуска приложения, например `..\bin\Debug\DynamicSchemes\`. Если все сделано правильно, там будет находиться файл RemarkGrafScheme.xml, на основании которого сформируется указанный пункт меню. Выбор этого пункта меню покажет граф «Замечание».

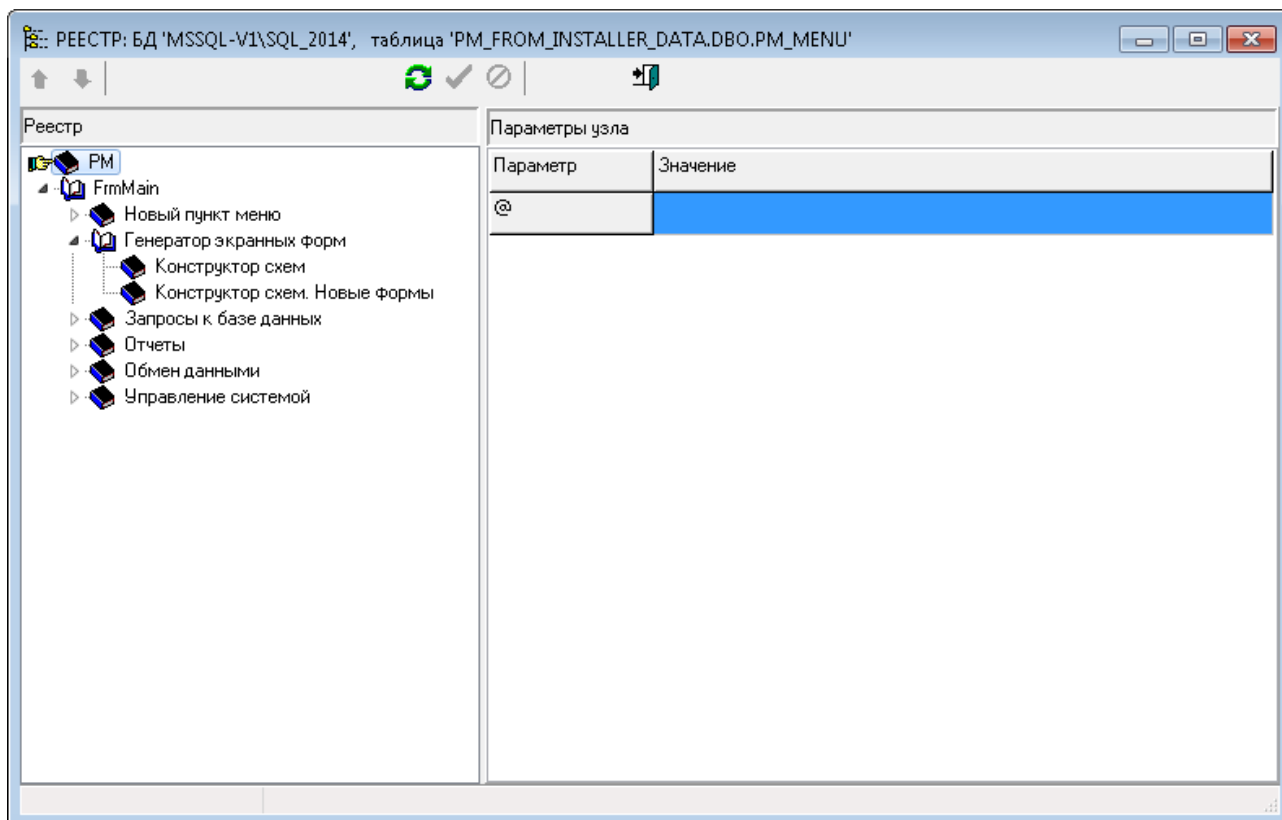
Это простейшей вариант вызова графа и, конечно, не всегда приемлемый.

Сценарий 2. Вызов формы-графа

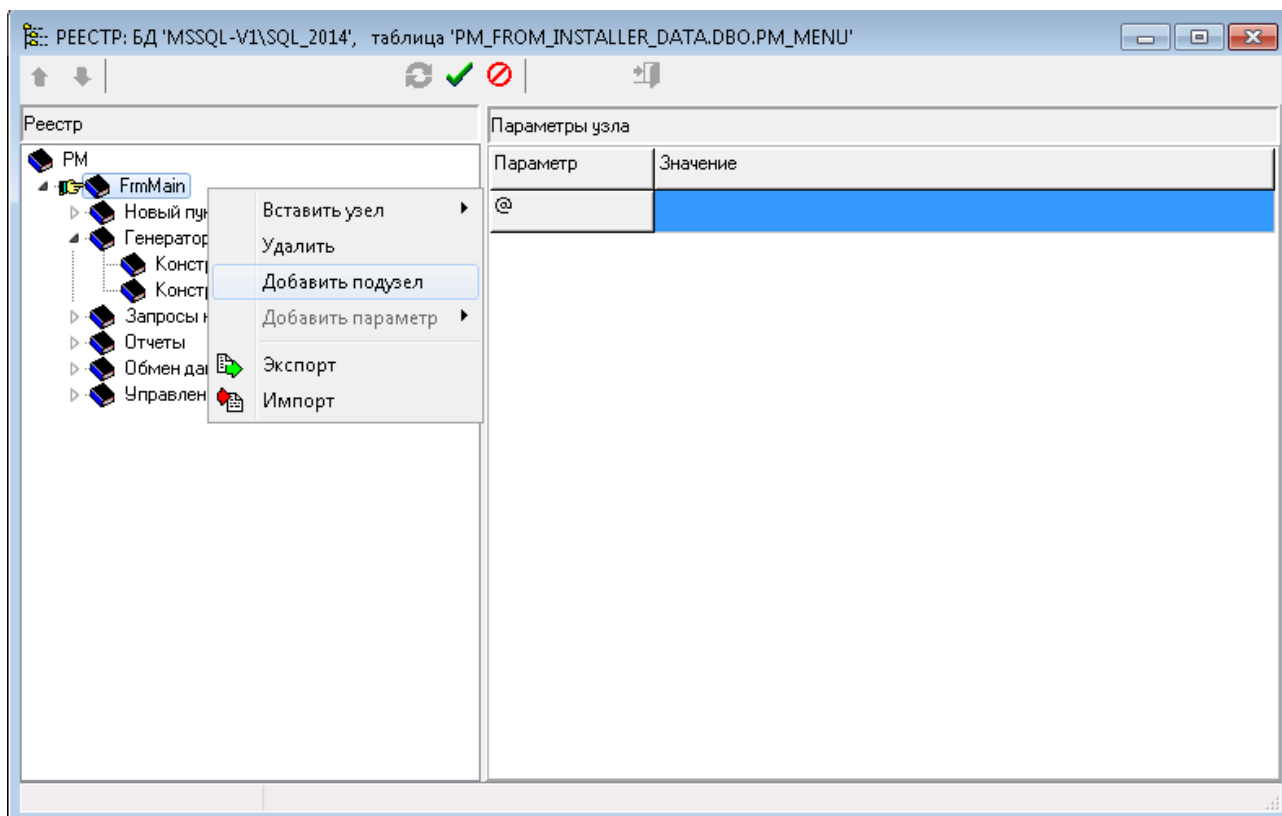
Для реализации вызова форма-графа вначале надо сохранить граф в виде VS-модуля и добавить его к проекту. Для этого можно использовать ту же кнопку конструктора форм «Добавить схему к проекту». Или можно воспользоваться, кнопкой «Сохранить XML-схему в файле», но в таком случае не будут автоматически созданы модули-формы для узлов-списков «Замечание» и «ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ». Возможно, формы для этих узлов-списков и не нужны (или они уже есть). Если не предполагается дополнительная функциональность, необходимые формы будут созданы автоматически при создании графа и специальные модули форм не нужны. Но рекомендуется модули-формы все-таки иметь отдельно, т.к. это дает возможность добавлять дополнительные кнопки и функционал на форму (см. [Шаг 6. Редактирование графа. Добавление пользовательской формы и кнопки на ней.](#)). Сохраненный модуль RemarkGrafScheme.cs требуется добавить к проекту вручную.

Для вызова графа можно добавить на главную форму FrmMain кнопку button1 и обработчик button1_Click, но рекомендуется использовать механизм *S_Технологии* **Меню**. Редактировать меню главной формы проекта нужно Утилитой правления проектом – **pjmgr.exe**, которая входит в состав пакета поставки. Настройка вызова утилиты и процесс добавления новых пунктов меню утилитой **pjmgr.exe** были описаны на Шаге 4, добавление вызова Справочника. Здесь действия аналогичные.

В пункте меню «Служебные таблицы» следует выбрать пункт «MENU».

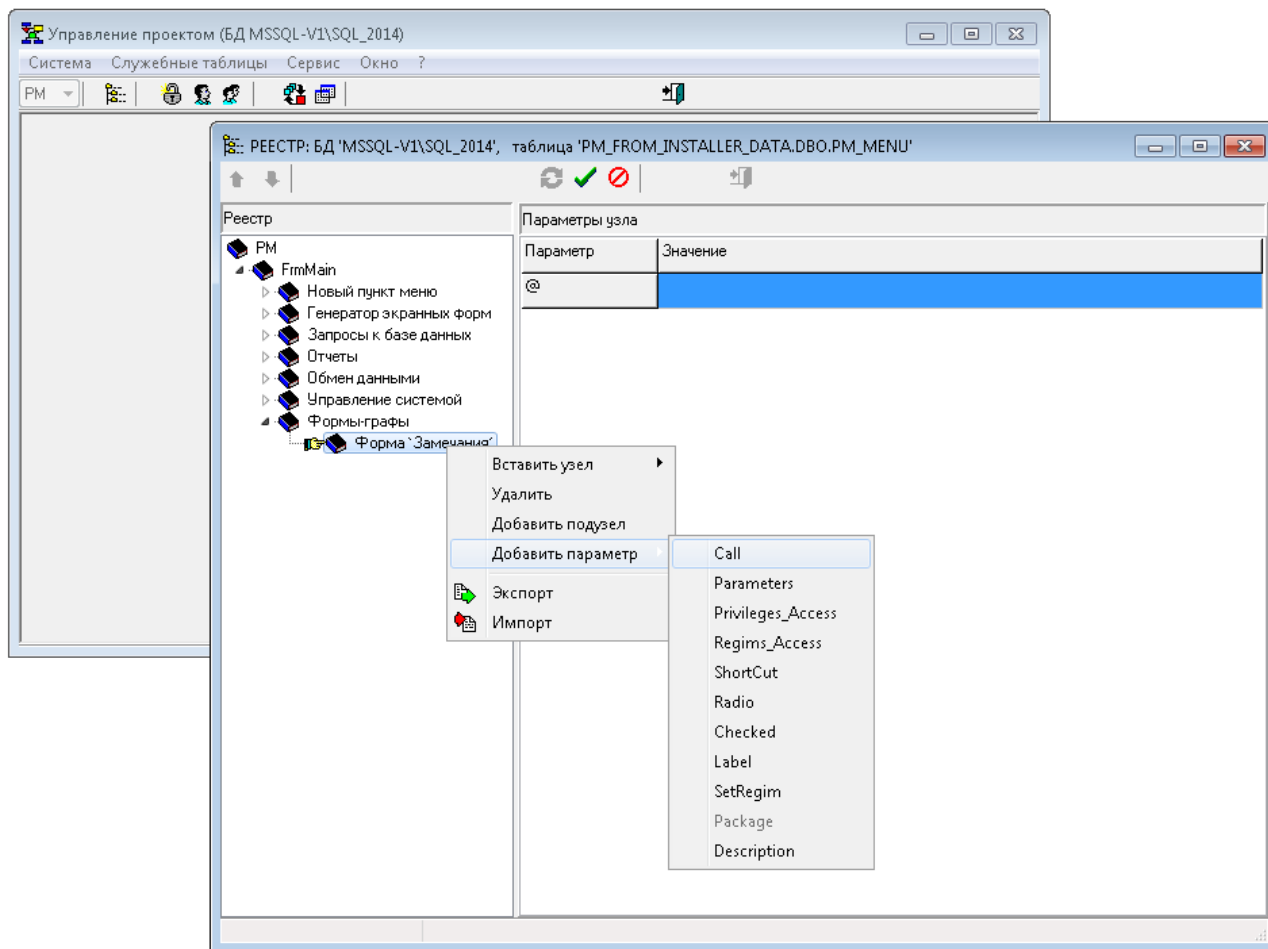


Затем нажать кнопку «Начать редактировать» и добавить новый узел, например, «Формы-графы». Для этого мышью перейти на корневой узел меню главной формы «FrmMain», правой клавишей мыши вызвать контекстное меню и выбрать пункт “Добавить подузел” :

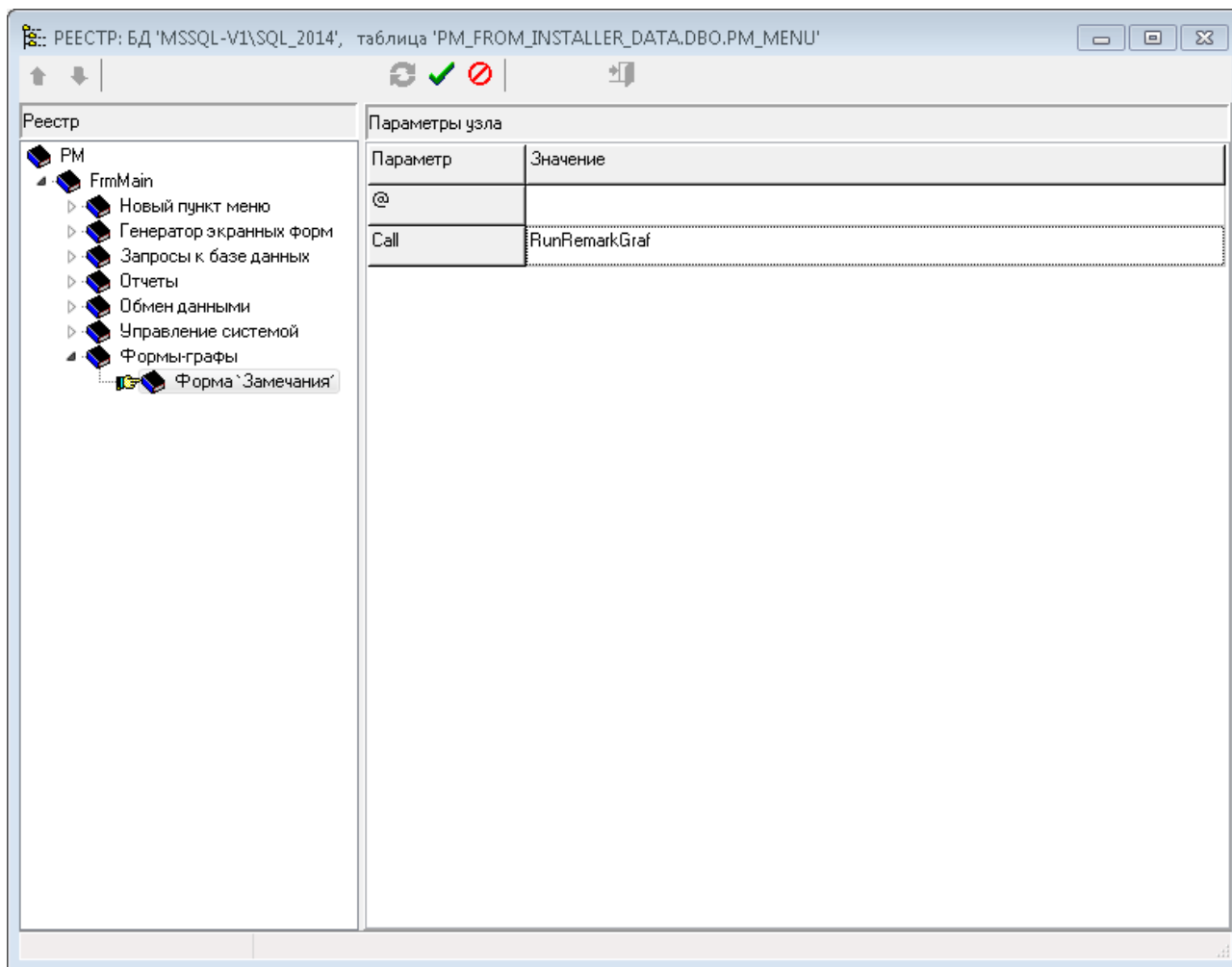


Далее требуется встать курсором на новый узел «Формы-графы» и добавить к нему подузел – «Форма 'Замечания'». Этот пункт меню и будет вызывать новый граф – «Замечание».

Для того, чтобы пункт меню вызывал какие-то действия, ему надо определить имя обработчика (в отличии от вызова класса-сущности как Справочника, где указывается просто имя Data-класса таблицы). Для определения обработчика нужно встать курсором на пункт «Форма 'Замечания'» и, вызвав контекстное меню, выбрать пункт «Добавить параметр» - «Call».



В качестве значения параметра ввести имя обработчика, например, RunRemarkGraf.



Задав имя обработчика, сохраняем изменения, завершаем работу с утилитой rjmgr.exe и возвращаемся к проекту.

Теперь нужно создать обработчик в модуле главной формы FrmMain.cs для добавленного пункта меню (или кнопки).

Для пункта меню нам потребуется обработчик с именем, определенным параметром Call пункта меню (см. выше, обработчик был определен как RunRemarkGraf). В FrmMain.cs нужно добавить следующий код:

```

/// <summary>
/// Форма 'Замечания'
/// </summary>
/// <param name="sender"></param>
/// <param name="parameters"></param>
public void RunRemarkGraf(ToolStripMenuItem sender, string parameters)
{
    WFGGraph.ShowGraph(RemarkGraf.SchemeText);
}

```

Тип метода-обработчика, реализующего пункт главного меню, должен быть именно такой. Далее нужно немного увеличить ширину формы, чтобы поместился вновь созданный горизонтальный пункт меню «Формы-графы».

Если используется кнопка, то вызов будет проще.

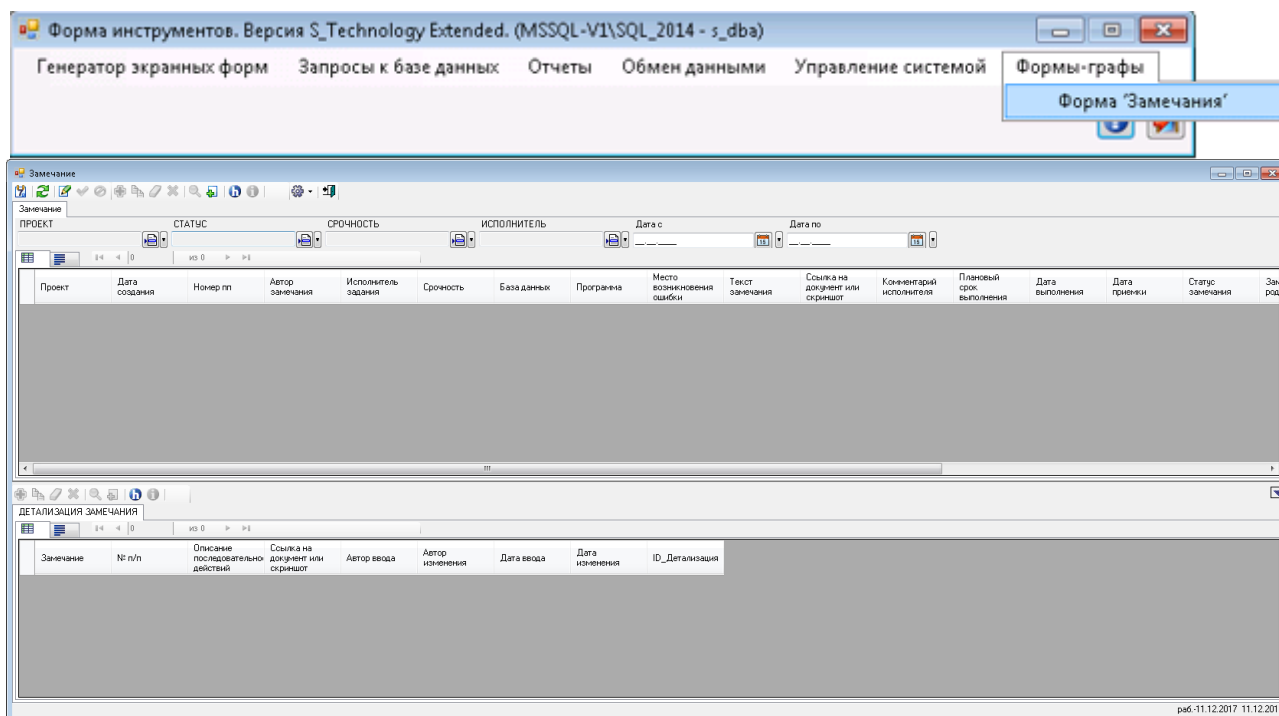
```

private void button1_Click(object sender, EventArgs e)
{
    WFGGraph.ShowGraph(RemarkGraf.SchemeText);
}

```

В обоих случаях для простого вызова графа используется метод `WfGraph.ShowGraph`, которому передается текст схемы графа. Возможны более сложные вызовы, тем же методом `WfGraph.ShowGraph` или создание графа вручную, с последующим переопределением свойств и визуализацией.

После пересборки и запуска проекта выбор нового пункта меню создаст и визуализирует созданный граф 'Замечание'.



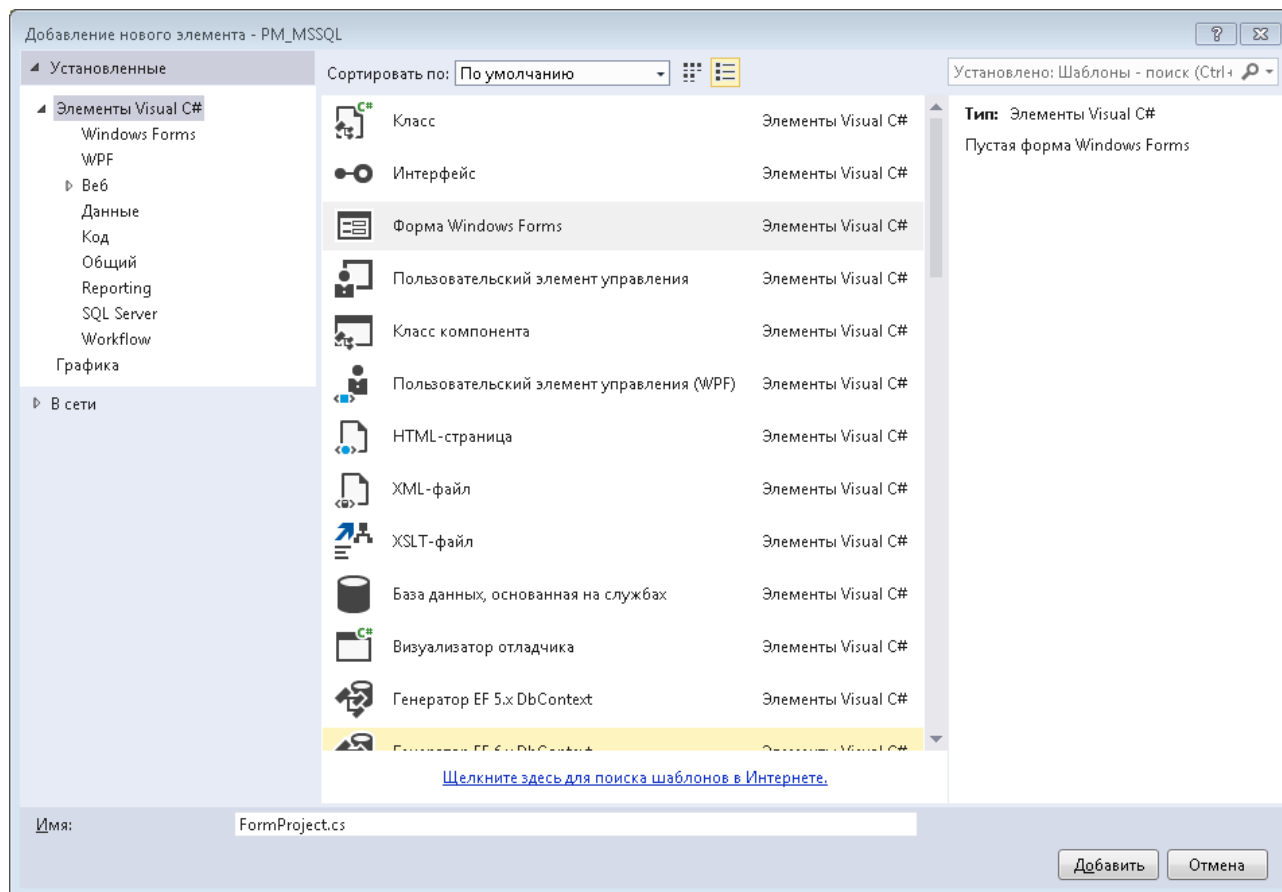
Используя возможности формы-графа, можно приступить к работе с таблицами.

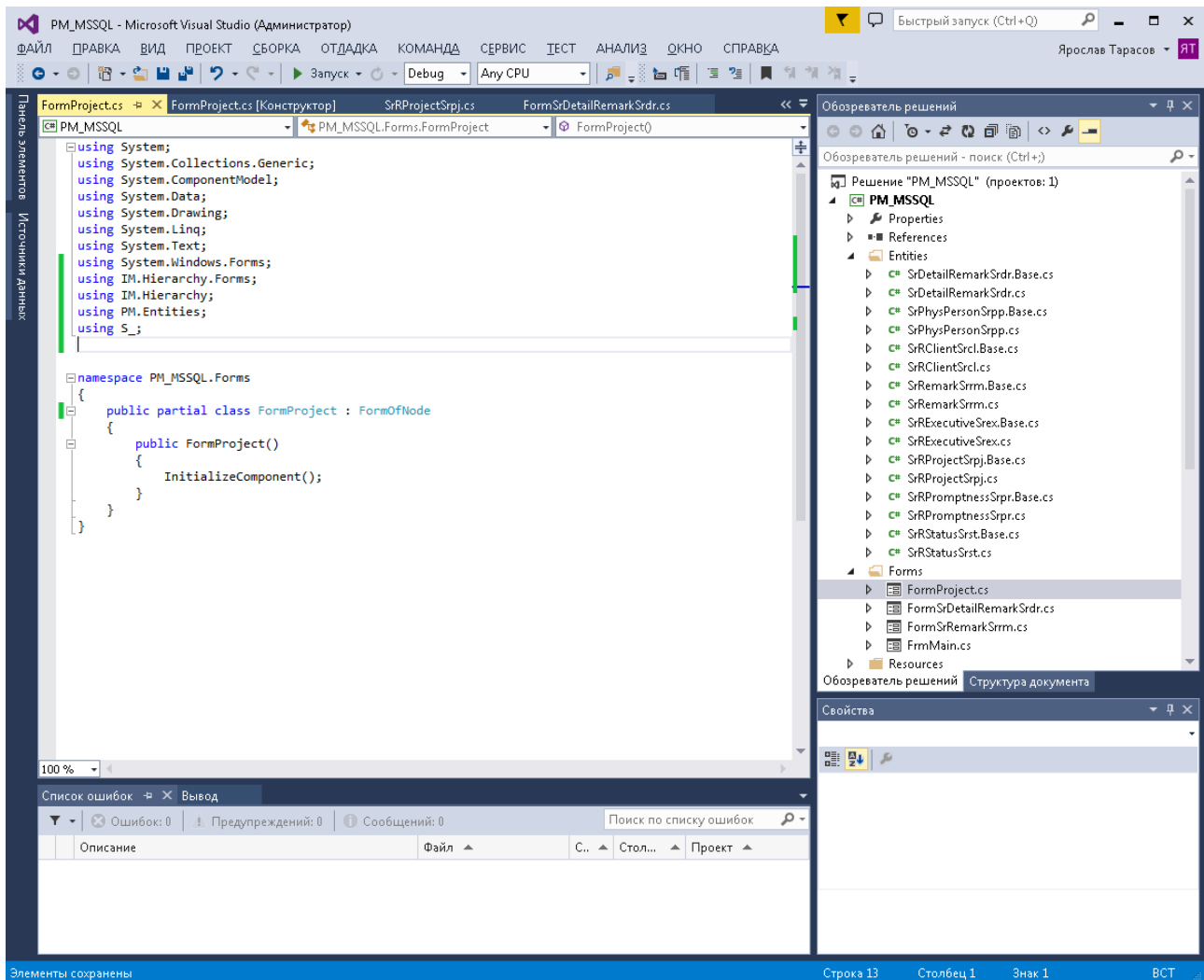
Шаг 6. Редактирование графа. Добавление пользовательской формы и кнопки на ней

Если для узла графа в описании указано значение свойства «Форма», то для такого узла при создании графа будет использована указанная форма. Для всех остальных узлов будет создана стандартная форма. В рассматриваемом примере узлы(сущности)-списки «Замечание» и «ДЕТАЛИЗАЦИЯ ЗАМЕЧАНИЯ» имеют определенные формы (`FormSrRemarkSrrm` и `FormSrDetailRemarkSrdr`), а узлы-фильтры («ПРОЕКТ» и «СТАТУС» и пр.) таких форм не имеют. Классы-формы `FormSrRemarkSrrm` и `FormSrDetailRemarkSrdr` создались и добавились в описание узлов-списков, потому что для сохранения описания использовалась кнопка «Добавить схему к проекту».

Зачастую функционала стандартных форм, которые автоматически создаются для узлов графа, не хватает и пользователю нужны какие-то действия, уникальные для конкретного узла (сущности). Например, нужна возможность к форме графа добавить пользовательские кнопки со специальным функционалом.

Для примера можно добавить кнопку на форму для сущности «ПРОЕКТ». Вначале нужно создать вручную форму для этой сущности, для чего следует добавить в папку Forms проекта новую форму и назвать ее, например, `FormProject`. В качестве предка формы нужно указать класс `FormOfNode`.





Далее нужно дополнить модуль формы следующим кодом.

```
using IM.Hierarchy.Forms;
using IM.Hierarchy;
using PM.Entities;
using S_;

namespace PM_MSSQL.Forms
{
    public partial class FormProject : FormOfNode
    {
        public FormProject()
        {
            InitializeComponent();
        }

        private SrRProjectSrpjEntity _EntityProject;

        /// <summary>
        /// Назначает сущность формы
        /// </summary>
        /// <returns>сущность формы</returns>
        public override s_EntityObject CreateEntity()
        {

```

```

        _EntityProject = new SrRProjectSrpjEntity();

        return _EntityProject;
    }

    /// <summary>
    /// Готовит форму к работе
    /// </summary>
    /// <returns>подготовлено успешно</returns>
    protected override bool s_Prepare()
    {
        if (!base.s_Prepare()) return false;

        if (View == null || !View.Prepare()) return false;

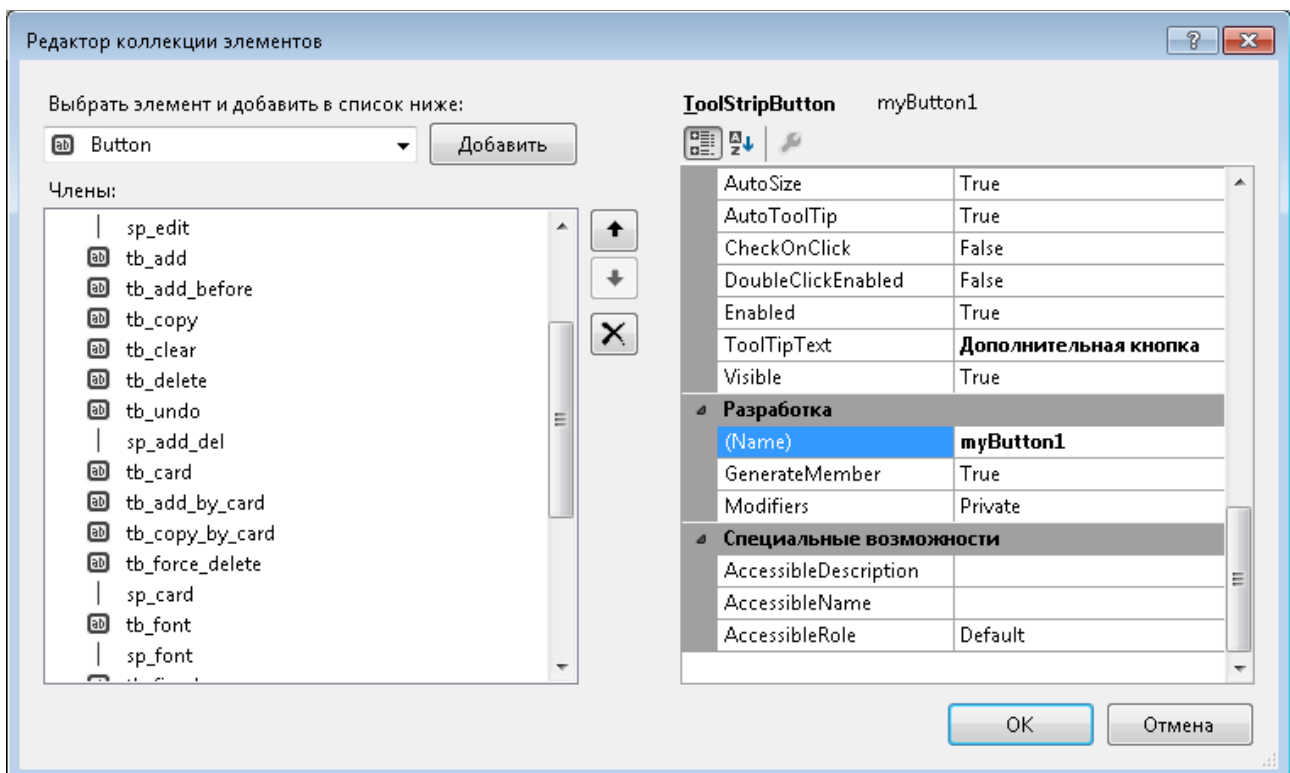
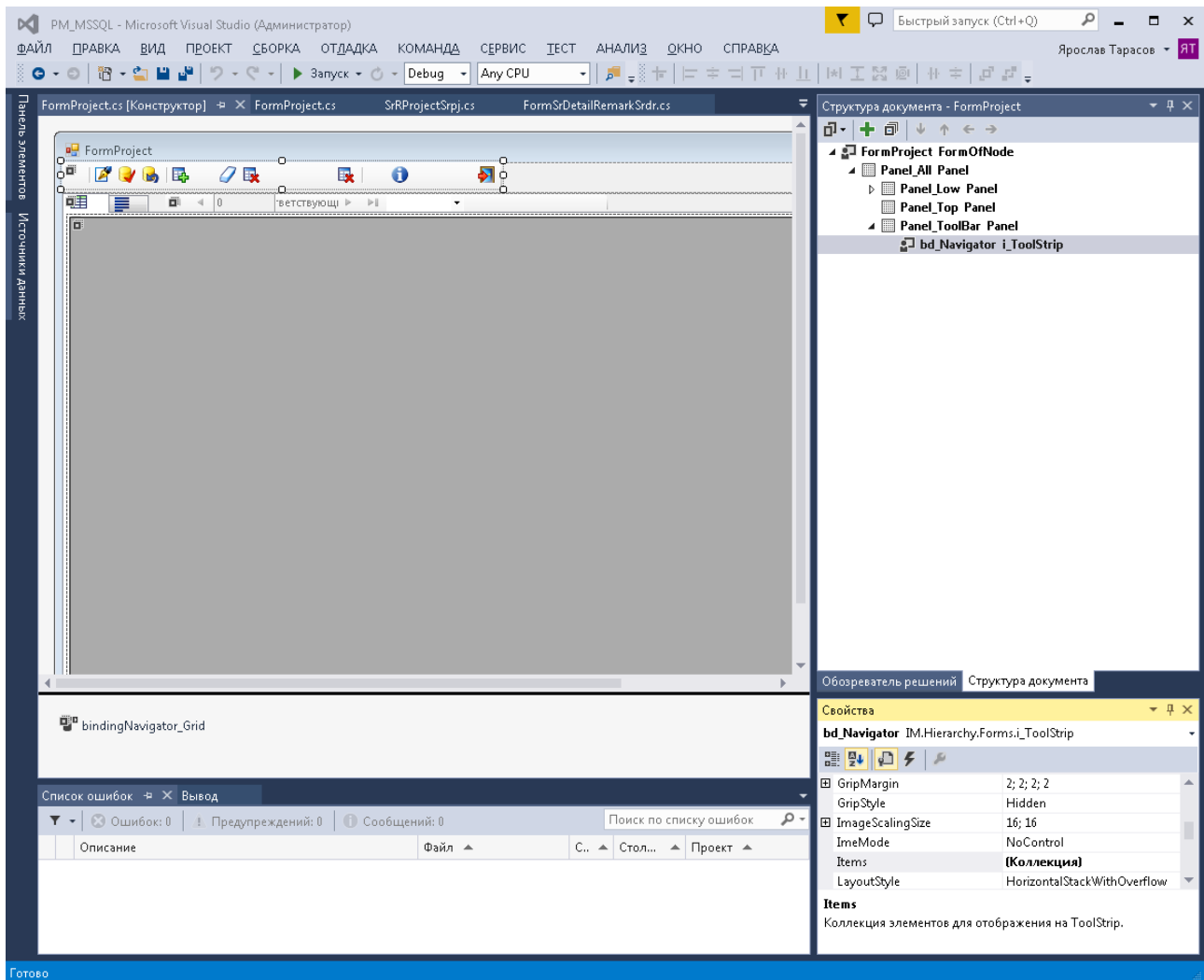
        return true;
    }

    /// <summary>
    /// Завершение работы формы. Удаляем сущность формы (_EntityProject).
    /// </summary>
    protected override void s_UnPrepare()
    {
        base.s_UnPrepare();

        if (_EntityProject != null)
        {
            _EntityProject.Dispose();
            _EntityProject = null;
        }
    }
}

```

Следующим шагом требуется добавить в дизайнера форм к `i_ToolStrip` `bd_Navigator` новую кнопку `myButton1`, и назначить ей иконку и обработчик нажатия.



Код модуля формы должен стать таким.

```
namespace PM_MSSQL.Forms
{
    public partial class FormProject : FormOfNode
    {
        public FormProject()
        {
            InitializeComponent();
        }

        private SrRProjectSrpjEntity _EntityProject;

        /// <summary>
        /// Назначает сущность формы
        /// </summary>
        /// <returns>сущность формы</returns>
        public override s_EntityObject CreateEntity()
        {
            _EntityProject = new SrRProjectSrpjEntity();

            return _EntityProject;
        }

        /// <summary>
        /// Готовит форму к работе
        /// </summary>
        /// <returns>подготовлено успешно</returns>
        protected override bool s_Prepare()
        {
            if (!base.s_Prepare()) return false;

            myButton1.Click += myButton1_Click; //Назначение обработчика для новой кнопки
            myButton1.Image = ImageList.Images["s_Light"]; //Назначение картинки для кнопки

            if (View == null || !View.Prepare()) return false;

            return true;
        }

        /// <summary>
        /// Обработчик нажатия новой кнопки
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        void myButton1_Click(object sender, EventArgs e)
        {
            s_MessageBox.Show("Hello");
        }

        /// <summary>
        /// Завершение работы формы. Удаляем сущность формы (_EntityTechnology).
        /// </summary>
        protected override void s_UnPrepare()
        {
            base.s_UnPrepare();

            if (_EntityProject != null)
            {
                _EntityProject.Dispose();
                _EntityProject = null;
            }
        }
    }
}
```



```

    }
}
}

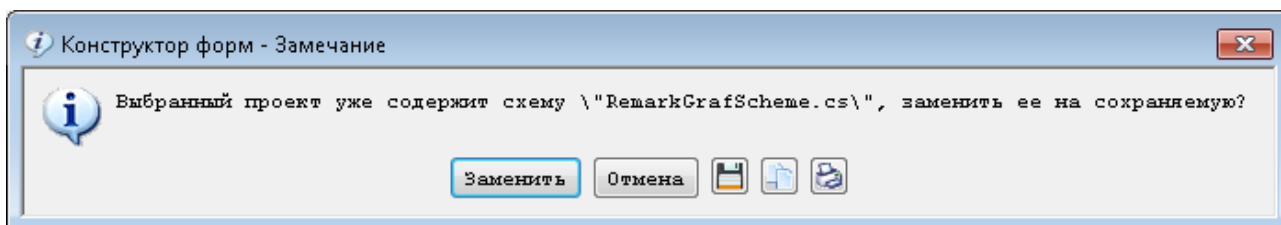
```

Теперь нужно изменить описание графа «Замечание». Для этого можно вызвать «Конструктор форм» другим способом, с формы-граф «Замечание», по специальной кнопке «Настройки».

После нажатия этой кнопки открывается форма «Конструктор форм» в несколько ином виде (режим «Схема»). При этом текущий граф «Замечание», уже загружен в форму. Можно перейти в более привычный вид – режим «Форма», нажав кнопку «Режим форма/схема», а можно оставаться в режиме «Схема». Для того чтобы внести изменения нужно включить редактирование.

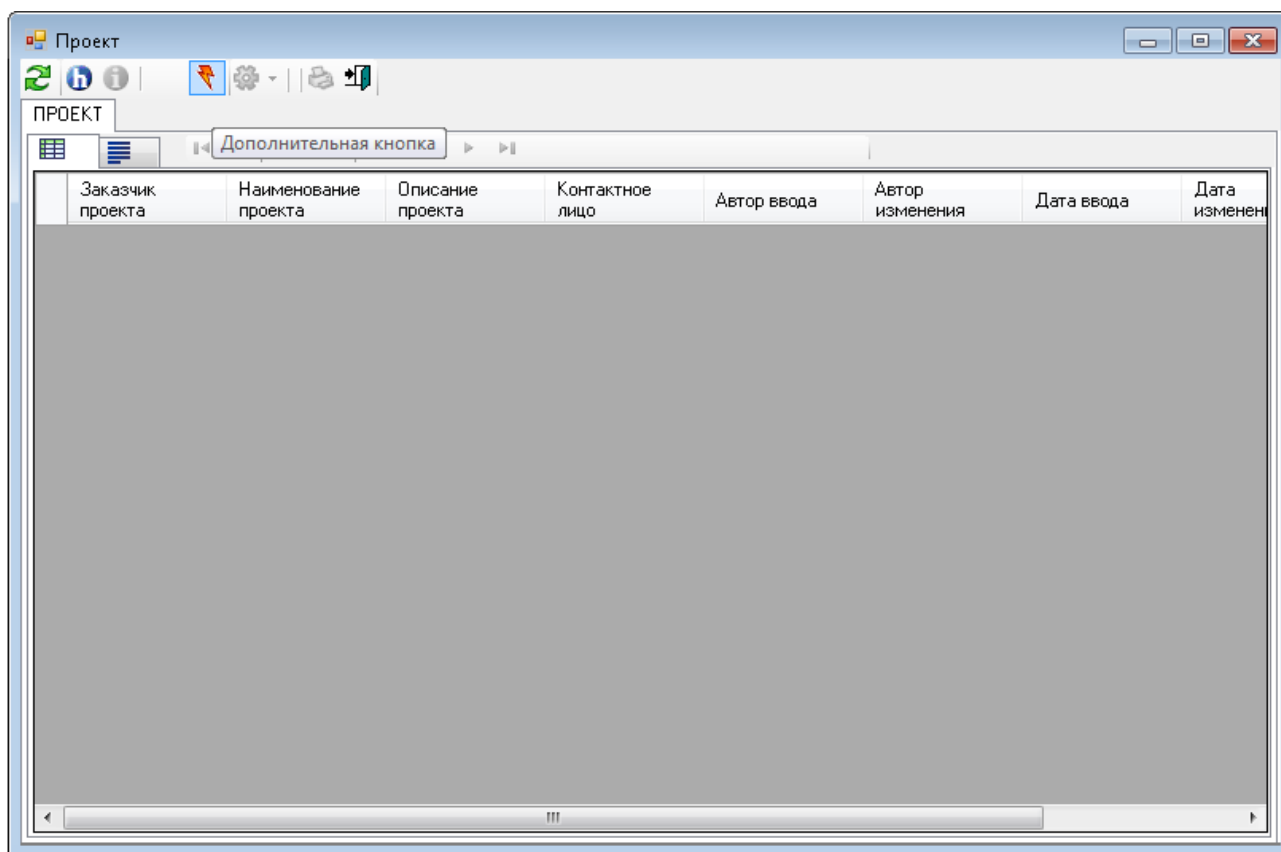
Имя класса формы (полное), по которому она будет создана. Если название формы не указано, то будет создана стандартная форма. Формат: namespace.type, assembly, например "MP.In_customers.MP". Класс сущности узла и его формы должны совпадать (Внутреннее имя: FormName)

Требуется изменить описания узла «ПРОЕКТ». Для этого нужно выбрать узел «ПРОЕКТ» и в свойстве «Форма» указать «PM_MSSQL.Forms.FormProject,PM» - имя класса формы с указанием пространства имен и, через запятую, «Имя сборки» проекта. Далее надо последовательно нажать кнопки «Сохранить» и «Добавить схему к проекту». Теперь информационное сообщение при сохранении будет более простое, предлагающее заменить уже имеющуюся в проекте схему.

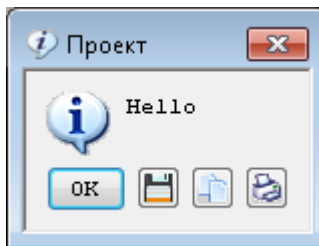


Нужно подтвердить изменения, нажав кнопку «Заменить».

После пересборки и повторного запуска проекта, в графе «Замечание» при открытии отбора «ПРОЕКТ» будет показана новая форма `FormProject` с дополнительной кнопкой.



При нажатии на эту кнопку должно появиться следующее окно.



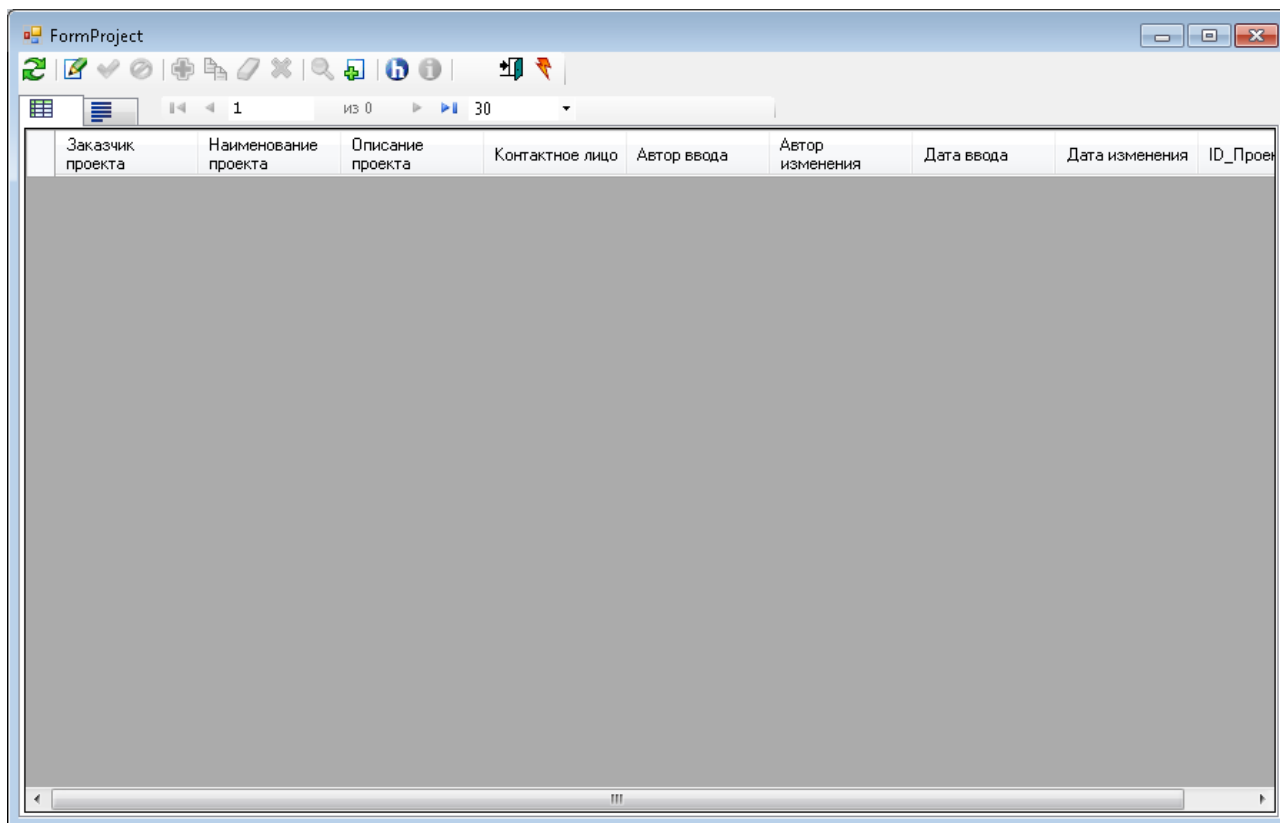
Выше было показано как можно изменить описание графа (подключить к узлу нестандартную форму), используя визуальный интерфейс «Конструктор форм». Можно изменить любые свойства «Графа», «Узлов», «Связей», корректируя табконтрол с таблицами свойств.

При определенном навыке и знании внутренних имен свойств (они видны в подсказке к каждому свойству в табконтрол свойств), несложные изменения (или даже сложные) в графе можно делать непосредственно корректируя xml-схему описания графа. Ниже приведено описание свойств, доступных для корректировки.

Например, изменение модуля RemarkGrafScheme.cs, класс `RemarkGraf`, свойство `SchemeText`. Можно было бы вручную добавить к узлу «ПРОЕКТ» новое свойство «`FormName=PM_MSSQL.Forms.FormProject,PM`».

```
...
<Node
  Caption                        = 'ПРОЕКТ'
  UID                           = '0'
  FormName                      = 'PM_MSSQL.Forms.FormProject,PM'
  EntityName                    = 'SrRProjectSrpjEntity'
  NameField                     = 'NAME_PROJ_SRPJ'
  KeyField                      = 'ID_SR_R_PROJECT_SRPJ'
  EnabledFilterTypes            = 'ft_show_data;ft_comboBox'
  PrivilegSuffix                = 'default'
  PrivilegViewName              = 'R_SR_R_PROJECT_SRPJ'
  PrivilegEditName              = 'W_SR_R_PROJECT_SRPJ'
  PrivilegInsertName            = 'W_SR_R_PROJECT_SRPJ'
  PrivilegDeleteName            = 'W_SR_R_PROJECT_SRPJ'
  PrivilegCopyName              = 'W_SR_R_PROJECT_SRPJ'
>
  </Node>
...
```

Созданную форму `FormProject` можно использовать и самостоятельно, а не в составе графа. Например, если добавить на главную форму приложения кнопку или пункт меню утилитой `pjmgr.exe`, и в обработчике написать следующий код `FormLow.Execute("PM_MSSQL.Forms.FormProject,PM")`, то при нажатии на эту кнопку (пункт меню) должна показаться форма следующего вида.



Заголовок формы в этом случае имеет значение `FormProject`, установленный при создании по умолчанию. Это происходит потому что форма вызвана вне графа и никто не установил ей привильный заголовок «ПРОЕКТ».

Далее приведено описание свойств «Графа», «Узла» и «Связи», доступных пользователю в конструкторе форм, которые можно устанавливать и изменять в табконтrole с таблицами свойств. Описываются их назначение и возможные значения.

1.1. Свойства графа

- «UID графа». UID графа нужен для отличия графов друг от друга. Например, в событиях, обслуживающих сразу несколько разных графов.
- «Режим описания». Режим, при котором обработчики нажатия элементов управления форм вместо выполнения действия, показывают описание этого действия. Предназначен для обучения конечного пользователя функциям интерфейса. Режим может быть включен непосредственно на форме графа через пункт всплывающего меню кнопки «Настройки».
- «Допустимы ли циклы в графе». Показывает допустимы ли циклы в графе. Или, другими словами, могут ли узлы графа ссылаться друг на друга по кругу.
- «Главная форма». Главная форма, на которой располагаются все узлы графа. По аналогии с формой узла графу может быть назначена форма, которую программист сделал вручную, создав потомка формы `FormOfGraph`. Такая необходимость может возникнуть, если программист желает добавить свои элементы управления на главную форму графа.
- «Разрешить смену ролей узлов». Разрешить пользователю менять роли узлов через всплывающее меню. Если каким-то узлам графа доступно более одной роли, можно разрешить конечному пользователю переключаться между ними (например, сделать узел, выступающий в роли мастер-сетки, отбором). Переключение

осуществляется через всплывающее меню узла или кнопки «Настройки» главной формы графа.

- «Сохранять настройки». Сохранять настройки, значения и роли узлов графа. При последующем запуске приложения граф должен восстановить сохраненные значения, т.е. должен выглядеть так, как он выглядел перед закрытием. Для разных пользователей приложения сохраняются свои настройки.
- «Индекс главного узла». Индекс узла, который будет установлен на роль главной сетки, при первом запуске приложения. Если этот параметр не указан, то на эту роль будет установлен любой из узлов, у которого доступна такая роль.
- «Скрывать сетки при выборе». Режим, при котором после выбора значения любого узла, все наборы данных подчиненных узлов скрываются и не показываются, пока не будет явной команды пользователя (например, нажатие кнопки «Обновить»). Также этот режим включается через всплывающее меню кнопки «Настройки».
- «Автовыравнивание отборов». Режим, при котором узлы в роли отбора (фильтра) будут размещаться автоматически в соответствии с их внутренними индексами. В противном случае узлы будут помещены в соответствии с номерами строки и столбца узла. Если программист желает сам размещать узлы, ему нужно отключить этот режим и проставить для каждого узла, выступающего в роли отбора, номер строки и столбца.
- «Радиус схемы». Радиус окружности, на которой размещаются узлы графа, когда он показывается в конструкторе форм.
- «Ширина отбора». Ширина по умолчанию панели, на которой создаются визуальные компоненты узла, выступающего в роли отбора.
- «Макс. отборов в строке». Максимальное число отборов в одной строке. Если число узлов в роли отбора больше этого числа, то будет начата новая строка, в которую будут помещены узлы, не поместившиеся в первой строке. Имеет смысл только если активен режим «Автовыравнивание отборов».
- «Автоконтроль ролей». Режим, при котором перед сменой ролей узлов происходит контроль корректности новых ролей. Если проверка не пройдена, то граф запрещает такую смену ролей узлов. По умолчанию включен.
- «Скрывать несвязанные нижние отборы». Режим, при котором при переключении между закладками, на которых лежат детальные узлы, остаются видимыми только те отборы, от которых зависит активная закладка.
- «Скрывать несвязанные верхние отборы». Режим, при котором при переключении между закладками, на которых лежат мастер-узлы, остаются видимыми только те отборы, от которых зависит активная закладка.
- «Автоматическая визуализация». Режим, при котором сам граф создает форму, служащую основой для форм всех узлов графа. В противном случае эту форму создает программист и назначает графу компоненты, на которые тот будет помещать узлы в той или иной роли. Например, если программист желает сделать сетки в три этажа, или разместить их на закладках своего TabControl'a.
- «Выбор периода». Отображать на главной форме кнопку выбора периода и сам период внизу главной формы.
- «Количество узлов». Отображает текущее количество узлов графа.
- «Количество строк на закладке». Количество строк на странице по умолчанию при включенном режиме листания в WEB-приложении.
- «Скрывать несвязанные детальные сетки». Режим, при котором при переключении между верхними закладками остаются видимыми только те детальные сетки, которые зависят от активной закладки.

- «Скрываемые кнопки». Скрываемые кнопки главной формы графа через ";". Важно понимать, что среди верхних кнопок управления на форме графа присутствуют как кнопки главной формы, так и кнопки, перенесенные из формы узла, являющегося активным среди мастер-узлов на данный момент, если их несколько.
- «Дерево». Граф является деревом, т.е. каждый узел имеет ровно одного родителя (за исключением корневого узла, у которого нет родителя). Все деревья являются графами, но не все графы являются деревьями.
- «Редактировать только в карточке». Режим, при котором пользователь может редактировать данные только в карточке (без редактирования в сетке). В этом случае пользователю недоступны кнопки «Начать редактирование», «Сохранить изменения», «Отказаться от изменений» на главной форме, а удаление записи нельзя отменить.
- «Привилегия на редактирование». Имя привилегии на редактирование тех узлов графа, у которых не указана ни одна из привилегий редактирования. Если указано это свойство, то для доступа к редактированию данных любого из узлов будет использовано это имя привилегии.

Замечание. Определение понятия привилегий, их добавление и проверка в программе см. ниже в разделе [Пользователь проекта. Система привилегий](#).

- «Соединение блоков». Соединение блоков: UID узлов, скобки и соединяющие фразы "and" или "or" (например, «(1 or 2) and 3»). Строка имеет смысл только для блочного режима графа, который включен для графов, созданных для конструктора запросов.
- «Два этажа на списках в карточке». В режиме «Карточка» сетки подчиненных узлов располагаются в два этажа, если это возможно.
- «Переносить условия графа в карточку». Переносить условия графа в карточку, чтобы ссылочные атрибуты карточки получили ограничения вызывающего графа. Как правило, сущность узла содержит несколько ссылочных атрибутов, связь с сущностями которых повторяет связь исходного графа. При включении этого режима, граф попытается наложить ограничения на набор данных, показываемый ссылочным атрибутом, если аналогичное ограничение есть в самом графе.
- «Проверять циклы в Prepare». Отложить проверку циклов до выполнения метода Prepare(). По умолчанию проверка происходит при каждом присоединении одного узла к другому узлу, что может сказаться на быстродействии. Имеет смысл, если в графе запрещены циклы.

1.2. Свойства узла

- «Название узла». Название узла, которое отображается над отбором или на закладке, соответствующей узлу.
- «UID узла». Уникальный идентификатор узла в рамках своего графа.
- «Значение узла». Значение узла (строка). Как правило, это ID выбранной записи (или нескольких записей через ";", если разрешен множественный выбор). Как правило, выбор значения осуществляется с помощью формы-списка, либо в ComboBox'e, либо с помощью формы, определенной пользователем.
- «Название выбранного объекта». Как правило, это название выбранного объекта, соответствующее значению узла.
- «Всегда очищать подчиненные узлы». После изменения значения узла, например, в результате выбора из формы, безусловно очищать значения во всех подчиненных узлах (на любую глубину подчиненности). В противном случае в подчиненных

узлах будет произведена проверка на соответствие их текущих значений новому значению в главном узле.

- «Форма». Имя класса формы (полное), по которому она будет создана. Если название формы не указано, то будет создана стандартная форма. Формат: namespace.type, assembly, например, «MP.fm_customers, MP». Класс сущности узла и его формы должны совпадать! Если это свойство не указано, граф сделает форму узла сам, создав потомка [FormOfNode](#).
- «Сущность». Имя класса сущности, по которому сущность узла будет создана. Если указана форма (только Windows Forms), то этот параметр не обязателен, и будет использована сущность формы.
- «Поле наименования». Имя атрибута сущности узла, из которого будет взято название выбранного значения узла.
- «Поле ID». Имя атрибута сущности узла, из набора данных которой будет браться значение узла. Допустимо указание нескольких полей через ";".
- «Доступные роли узла». Роли узла через ";", на которые он может переключаться во время работы графа.
- «Доступные типы отбора». Типы отбора узла через ";", на которые он может переключаться, если узел выступает в роли отбора.
- «Разрешен множественный выбор». При выборе значения узла в открывшейся форме будет доступна возможность выбора более одного объекта через карман.
- «Обязателен к заполнению». Узел обязателен к заполнению, все узлы в виде списков не смогут показать данные, пока значение этого узла не будет определено.
- «Дополнительное условие». Текст дополнительного условия, которое будет добавляться к тексту SQL-запроса открываемого набора данных.
- «Текст SQL-запроса». Текст SQL-запроса, который будет использован для построения набора данных узла. Этот параметр используется только в качестве альтернативы указанию имени сущности. В этом случае пользователь не сможет редактировать набор данных узла.
- «Отображать зависимость в момент выбора». Режим, при котором в момент выбора нового значения узла из формы, во всех остальных узлах сразу отображается зависимость от активной записи формы, хотя выбор еще не был произведен.
- «Текст пустого отбора». Текст, который будет показан в узле, играющем роль отбора (фильтра), если значение узла не выбрано (например, «Все записи»).
- «Разрешить внешние таблицы». Разрешить использовать внешние таблицы (ORACLE, MSSQL) для хранения значения узла, если разрешен множественный выбор значения узла. Параметр предусмотрен для преодоления ограничения на количество ключей в секции IN SQL-запроса.
- «Суффикс». Суффикс, который используется для формирования стандартных имен привилегий, если их имена не указаны явно. Как правило, указывается суффикс из таблицы сущности. Например, если задать суффикс «POAA», то граф будет проверять наличие у пользователя привилегий «VIEW_POAA» на просмотр, «CHAN_POAA» на редактирование, «BUILD_POAA» на добавление, «DEL_POAA» на удаление и «COPY_POAA» на копирование. Более подробно о использовании привилегий см. раздел [Пользователь проекта. Система привилегий](#).
- «Привилегия на просмотр». Имя привилегии на просмотр данных узла.
- «Привилегия на добавление». Имя привилегии на добавление новых записей.
- «Привилегия на удаление». Имя привилегии на удаление записей.
- «Привилегия на копирование». Имя привилегии на копирование записей.

- «Внешняя схема». Имя класса XML-схемы, которая будет использована для выбора значения узла. В момент выбора будет создан экземпляр графа по указанной схеме, который будет показан в режиме выбора одной или нескольких записей. Класс внешней схемы должен поддерживать интерфейс [IScheme](#). Например, «IM.Hierarchy.RemarkGraf, PM».
- «Индекс узла во внешней схеме». Индекс узла во внешней схеме, выступающего в роли мастер-сетки, из которого будет выбрано значение для текущего узла. Как правило, имена сущностей узлов совпадают.
- «Не переносить значения узлов во внешнюю схему». Не переносить значения узлов во внешнюю схему и не блокировать их. Если у узла указана «внешняя схема», то граф не будет искать среди узлов внешнего графа узлы, похожие на свои, и не будет переносить туда значения из аналогичных узлов, а также не будет блокировать к ним доступ.
- «№ строки». Номер строки, в которую будет помещен узел, если ему будет выбрана роль отбора (фильтра). Нумерация начинается с 1. Имеет смысл только при отключенном режиме графа «Автовыравнивание отборов».
- «№ колонки». Номер колонки, в которую будет помещен узел, если ему будет выбрана роль отбора (фильтра). Нумерация начинается с 1. Имеет смысл только при отключенном режиме графа «Автовыравнивание отборов».
- «№ закладки». Номер закладки, в которую будет помещен узел, если ему будет выбрана роль списка. Нумерация начинается с 1.
- «Объединять атрибуты». Объединять Custom-атрибуты, явно указанные в XML-схеме, с атрибутами, полученными из сущности. Если в схеме для узла указаны Custom-атрибуты с какими-то свойствами, то граф попытается объединить эти свойства со свойствами одноименных атрибутов сущности узла (например, свойства «Видимость», «Название атрибута»).
- «Только для чтения». Только для чтения, редактирование набора данных узла запрещено.
- «Недоступен пользователю». Если узел в виде отбора, то он недоступен для выбора, если в виде формы – редактирование в сетке запрещено. Т.е. недоступен для взаимодействия с пользователем.
- «Скрываемые кнопки». Список скрываемых кнопок формы через ";".
- «Показываемые кнопки». Список показываемых кнопок формы через ";".
- «Разделитель между значениями». Разделитель между значениями узла. Значения хранятся одной строкой, одно значение отделяется от другого этим разделителем. По умолчанию разделитель ",".
- «Не сохранять значение». Не сохранять значение узла при включенном режиме графа «Сохранять настройки».
- «Разрешено ли листание». Разрешено ли постраничное отображение набора данных узла. При включении этого режима набор данных узла будет показываться порциями, а не целиком.
- «Редактировать только в карточке». Редактировать только в карточке (без редактирования в сетке). Аналогично одноименному режиму графа.
- «Дополнительный java-скрипт». Путь к дополнительному java-скрипту, который дополняет поведение html-тегов, нарисованных для текущего узла (пример: "/win/css/jquery.window.css"). Только для WEB'а.
- «Разделитель между полями». Разделитель между полями. Если в свойстве узла «Поле ID» указано несколько полей, то разделяем значение каждого поля этим разделителем. Например, если указано два поля, значение узла примет вид

«ID1_первого_поля; ID1_второго_поля, ID2_первого_поля; ID2_второго_поля,[и т.д.]».

- «Разрешить отметку строк». Разрешить отметку строк. Строки выборки могут отмечаться чекбоксами.
- «Роль в схеме». Какую роль в схеме играет узел: узел схемы, часть блокового условия, часть результата. Только для блокового режима графа, используемого в конструкторе запросов.
- «Имя таблицы». Имя таблицы БД, если для узла не указана сущность. Используется в блоковом режиме графа.
- «SelectionName дата-класса». Имя выборки дата-класса сущности узла. Используется чтобы граф создавал дата-класс сущности не с именем выборки «Default», а с указанной программистом.

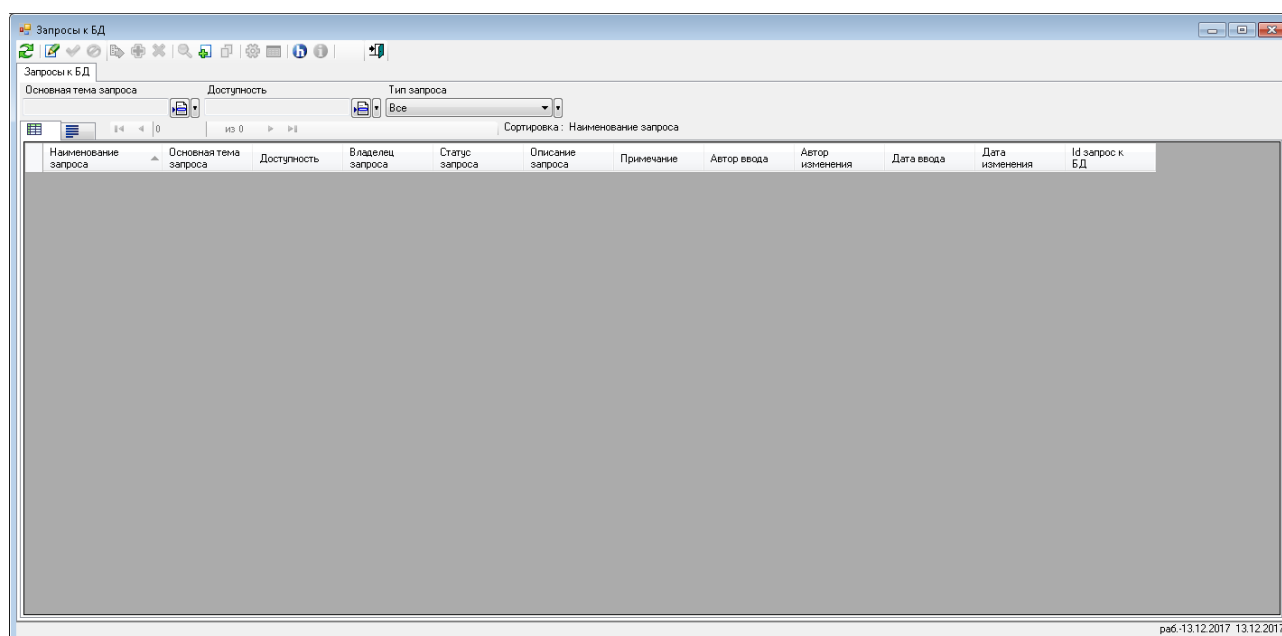
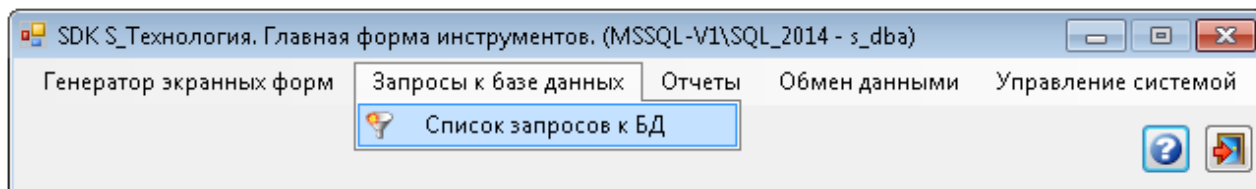
1.3. Свойства связи

- «UID главного узла». UID главного узла в связи "главный-подчиненный".
- «UID дочернего узла». UID дочернего узла в связи "главный-подчиненный".
- «Условие связи». Текст, который будет добавлен к тексту SQL-запроса подчиненного узла, если в главном узле выбрано непустое значение. В тексте, как правило, указывается параметр :ID, вместо которого будет подставлено значение главного узла.
- «Ссылочное поле». Имя поля в подчиненном узле, через которое осуществляется ссылка на главный узел. Если этот параметр указан, то при добавлении новой записи в набор данных подчиненного узла в это поле будет автоматически вставлено значение из главного узла. Кроме того, этот параметр обязателен при включенном режиме связи «Применять после открытия».
- «Главный узел обязателен». Набор данных дочернего узла не будет открыт до тех пор, пока не будет выбрано значение в главном узле.
- «Применять после открытия». Режим, при котором фильтрация набора данных подчиненного узла осуществляется не через изменение фразы Where запроса, а через свойство Filter, что работает медленнее и занимает больше памяти.
- «Скрываемые колонки». Атрибуты подчиненного узла, которые будут скрыты в случае, если значение главного узла не пусто и будут показаны, если пусто.
- «Показываемые колонки». Атрибуты подчиненного узла, которые будут показаны в случае, если значение главного узла не пусто и будут скрыты, если пусто.
- «Применять всегда». Режим, при котором связь будет отрабатываться всегда, в каких бы ролях ни выступали связываемые узлы.
- «Тип связи». Тип связи узлов между собой. Это либо SQL-запрос целиком, либо выбор из списка, либо конкретная операция (равно, не равно и т.п.). Имеет смысл только для блокового режима графа.
- «Способ подключения правой части». Способ подключения правой части связи между узлами. Это либо выбор из списка (фильтр на сетке), либо константа, либо поле таблицы. Имеет смысл только для блокового режима графа.
- «Добавлять апострофы». Добавлять апострофы к значению мастер-узла.

Шаг 7. Инструмент «Конструктор запросов»

Помимо возможности создавать xml-схемы графов, «Конструктор форм», рассмотренный выше, обладает функционалом по созданию запросов к БД.

Для того чтобы запустить конструктор форм в режиме конструктора запросов нужно открыть форму списка запросов к БД.



Добавить новый запрос (заголовок запроса).

Наименование запроса	Основная тема запроса	Доступность	Владелец запроса	Статус запроса	Описание запроса	Примечание	Автор ввода	Автор изменения	Дата ввода	Дата изме
Единицы измерения	С_ЕДИНИЦА ИЗМЕРЕНИЯ	Лично	s_dba	В подготовке	Единицы измерения типа Время		S_DBA		13.12.2017 11:59	

После заполнения полей с описанием запроса нужно нажать на кнопку «Конструктор запросов». Откроется форма, похожая на «**Конструктора форм**», но с некоторыми отличиями. На этой форме можно ввести «тело» запроса, для этого следует заполнить панели Сущностей для запросов, Условий запросов и Выборки.

С_ЕДИНИЦА ИЗМЕРЕНИЯ

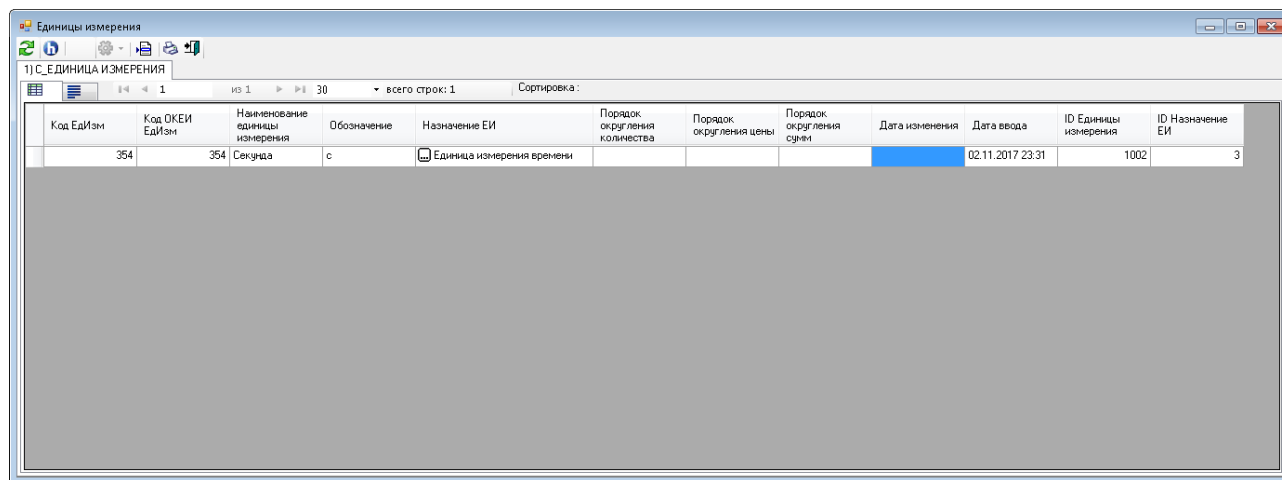
Блок условий №1 С_ЕДИНИЦА ИЗМЕРЕНИЯ.ID Назначение ЕИ = константа 3

1) С_ЕДИНИЦА ИЗМЕРЕНИЯ

Это пример простейшего запроса для выборки из таблицы «Единицы измерения» (S_EI_EIEI) записей, имеющих «Назначение» - «Единицы измерения времени» (ID_R_MEASURE_DS_EIEI=3). Более подробно шаги проектирования запроса см. в отдельном документе “ИМ Конструктор запросов.doc”.

Готовый запрос следует сохранить и можно сразу его выполнить для проверки, используя кнопку «Показать форму на основе XML-схемы».

Сохраненный запрос может быть выполнен в любой момент с формы списка запросов по кнопке “Выполнить запрос”. И результат будет показан в привычной форме.



Код ЕдИзм	Код ОКЕИ ЕдИзм	Наименование единицы измерения	Обозначение	Назначение ЕИ	Порядок округления количества	Порядок округления цены	Порядок округления сумм	Дата изменения	Дата ввода	ID Единицы измерения	ID Назначение ЕИ
354	354	Секунда	с	Единица измерения времени					02.11.2017 23:31	1002	3

Это очень поверхностное и схематичное описание инструмента «Конструктор запросов». Более подробное описание см. в документе “ИМ Конструктор запросов.doc”.

Шаг 8. Создание нового проекта и его переименование. Дальнейшее развитие приложения

Создание проекта с произвольным именем сборки

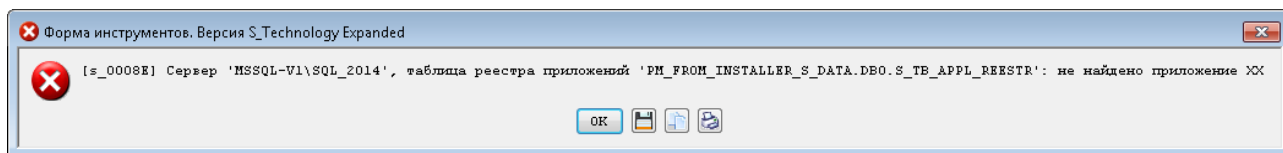
До сих пор демонстрировалась работа исключительно с проектом, который был создан с помощью расширения Microsoft Visual Studio «IMProjectWizard». Это расширение, с помощью Конфигуратора, создает БД с необходимыми таблицами и VS-проект с произвольным именем (в этом документе это проект PM_MSSQL.csproj), но со строго определенным именем сборки – “**PM**”. Т.е. приложение всегда носит имя **PM.exe**.

Это сделано не случайно. Дело в том, что у S_Технологии существует понятие «Реестра приложений». Физически это таблица БД S_TB_APPL_REESTR, в которой хранится особая служебная информация, и данные приложения PM занесены в этот реестр.

В поставку включено средство, позволяющее увидеть и изменить «Реестр приложений», это уже знакомая утилита **pjmgr.exe**. С помощью ее на Шагах 4 и 6 изменялось Меню главной формы. С помощью этой же утилиты можно просмотреть и изменить «Реестр приложений».

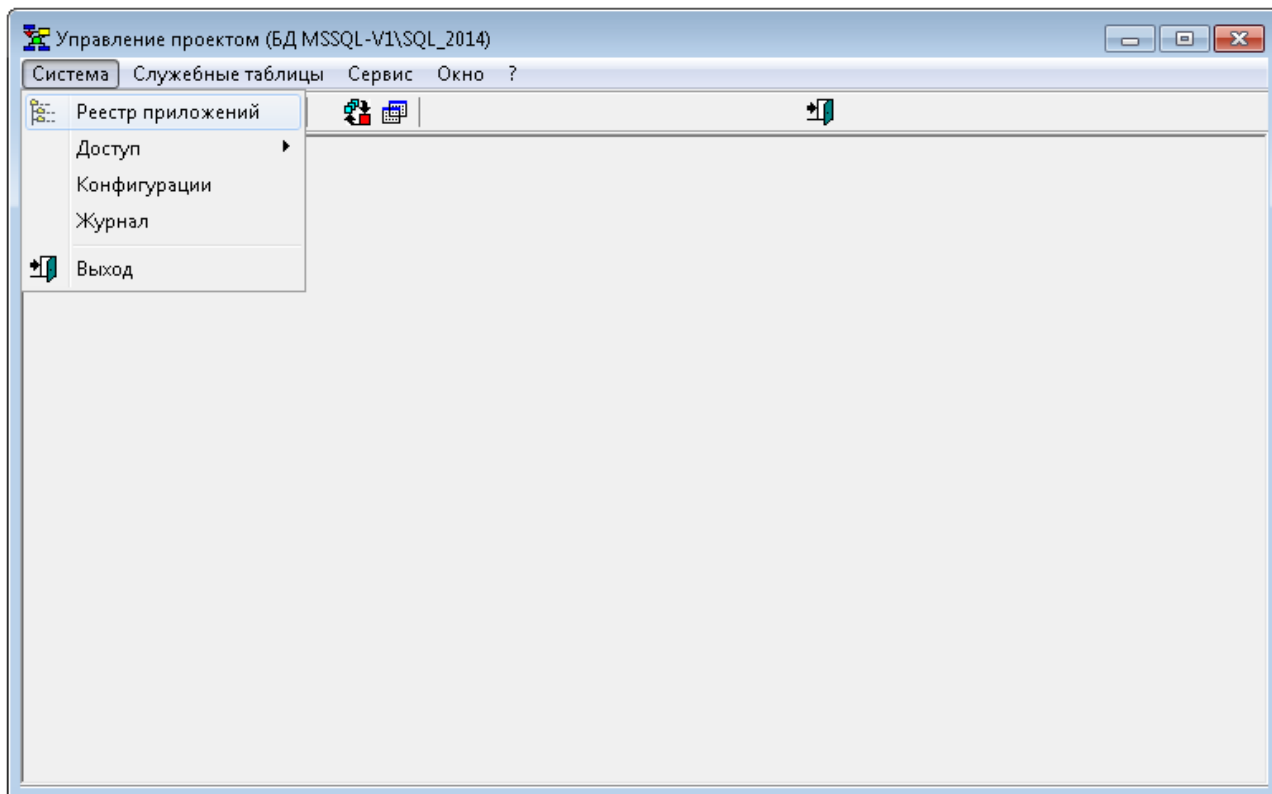
Изменить может быть необходимо потому, что фиксированное имя **PM.exe** не всегда подходит для любого проекта. Программист может захотеть назвать сборку, исходя из специфики конкретного проекта, к примеру сборка “XX_15_Z4.exe”. Или другое имя сборки может понадобиться если создается не первое приложение, и имя сборки PM.exe уже занято.

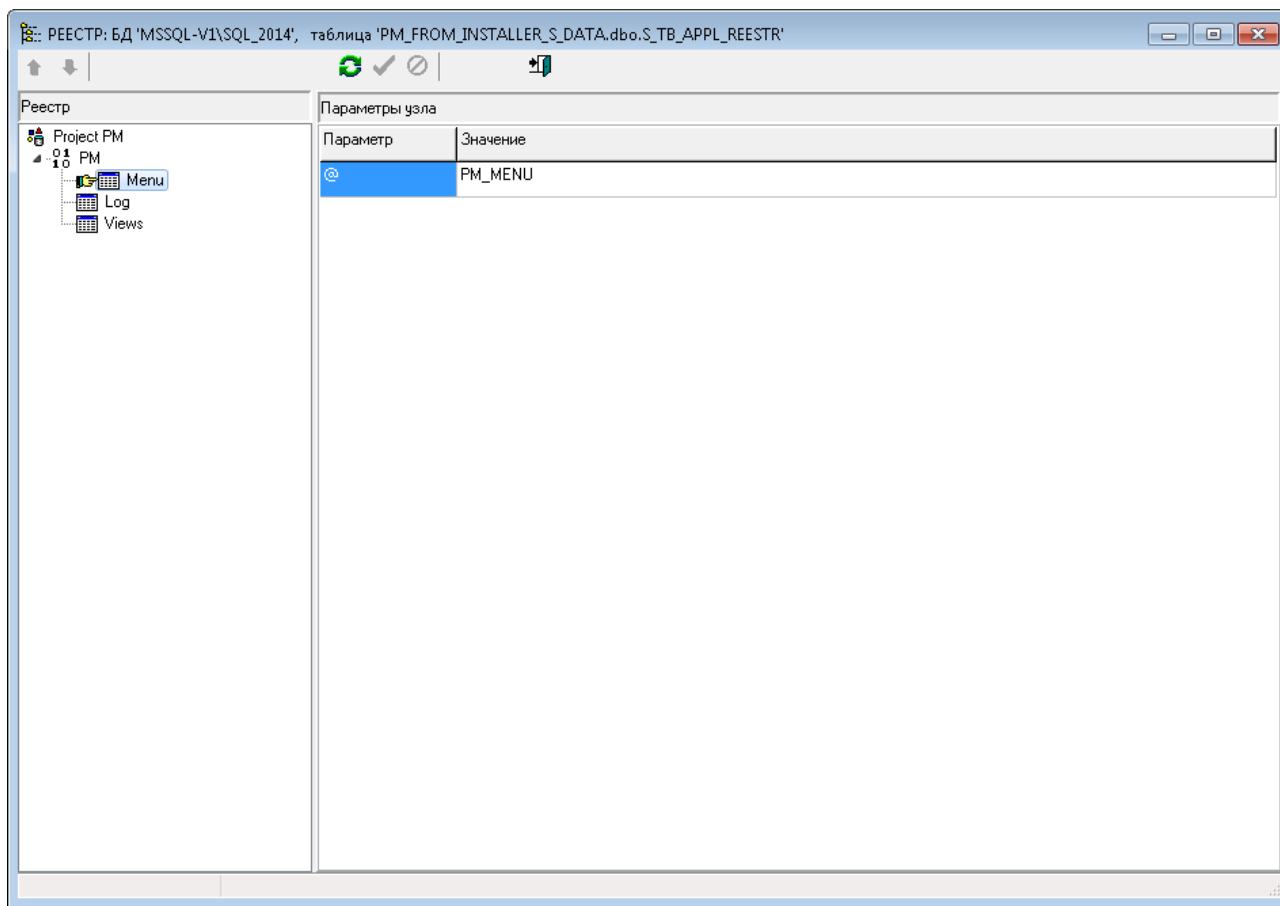
Если мы просто возьмем и изменим имя сборки, при запуске приложения возникнет ошибка.



Ошибка возникает потому, что *S_Technologia* по двум первым буквам имени сборки пытается найти приложение в «Реестре приложений» и воспользоваться его описанием.

«Реестр приложений», созданный Конфигуратором при создании проекта, можно открыть с помощью утилиты **pjmgr.exe**.





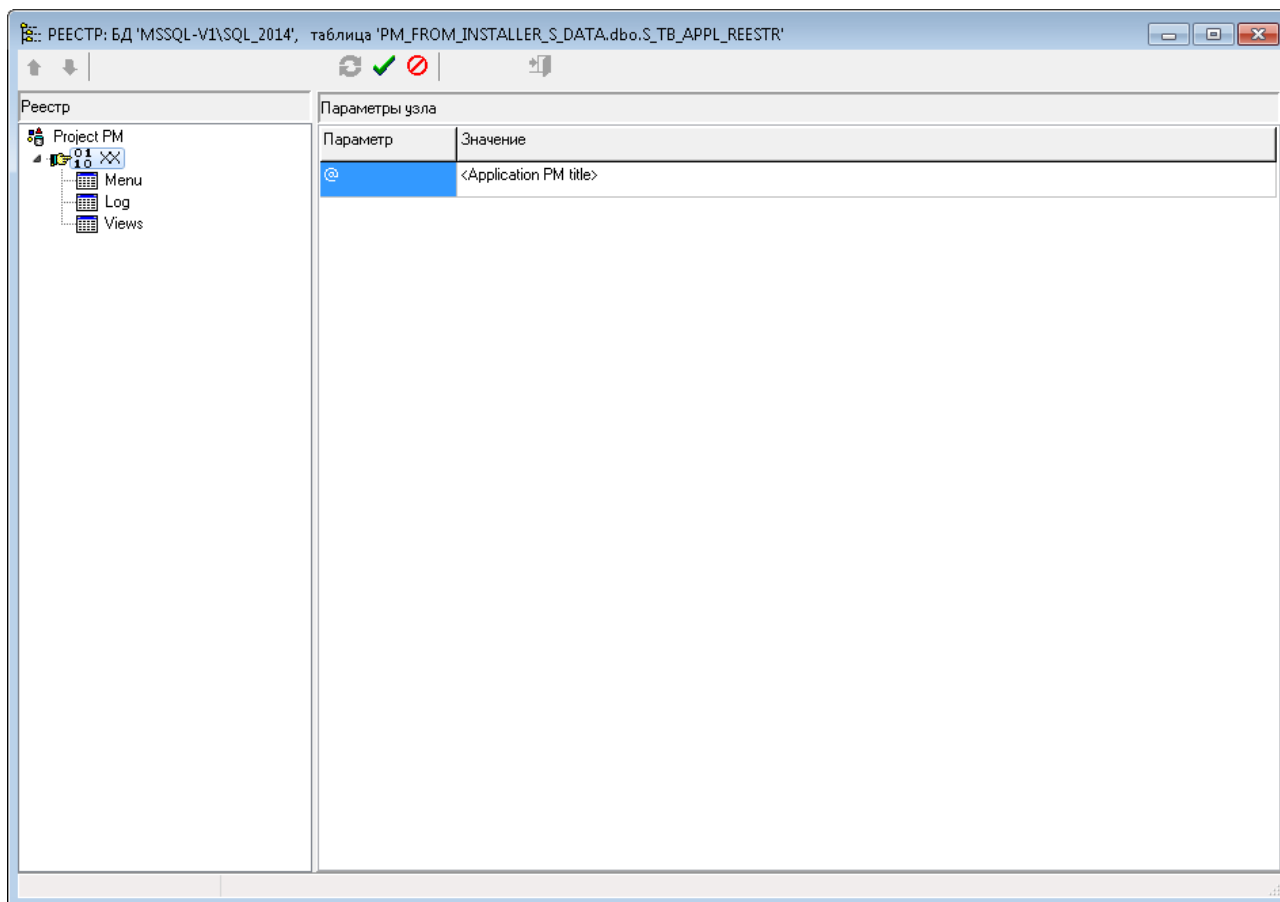
«Реестр приложений» имеет иерархическую структуру.

1. Первый уровень – «Проект». Логическое понятие *S_Технологии*. Объединяет одно или несколько «Приложений». В данном примере «Реестре приложений» имеет имя «Project PM». Это имя произвольное.
2. Второй уровень – «Приложения». В данном примере в «Реестре приложений» будет одно приложение с именем «**PM**». Именно с этим именем должны совпадать первые два символа имени сборки VS-проекта (exe-файла).
3. Третий уровень – «Описание служебных таблиц Приложения». В Приложении «**PM**» описаны 3 таблицы:
 - Menu – физическое имя таблицы PM_MENU, содержит описание меню главной формы.
 - Log – физическое имя таблицы PM_LOG, содержит Журнал системы.
 - Views – физическое имя таблицы PM_VIEWS, содержит описание служебных Views приложения.

У «Реестра приложений» много функций, в том числе и определение служебных таблиц «Приложений» - PM_MENU, PM_LOG и PM_VIEWS. Здесь требуется жесткая привязка имени «Приложения» к имени сборки VS-проекта (exe-файла), чтобы exe-файла нашел именно свои таблицы Menu, Log и Views. Поэтому также строго определяется имя сборки – “**PM**” при создании проекта из расширения «IMProjectWizard».

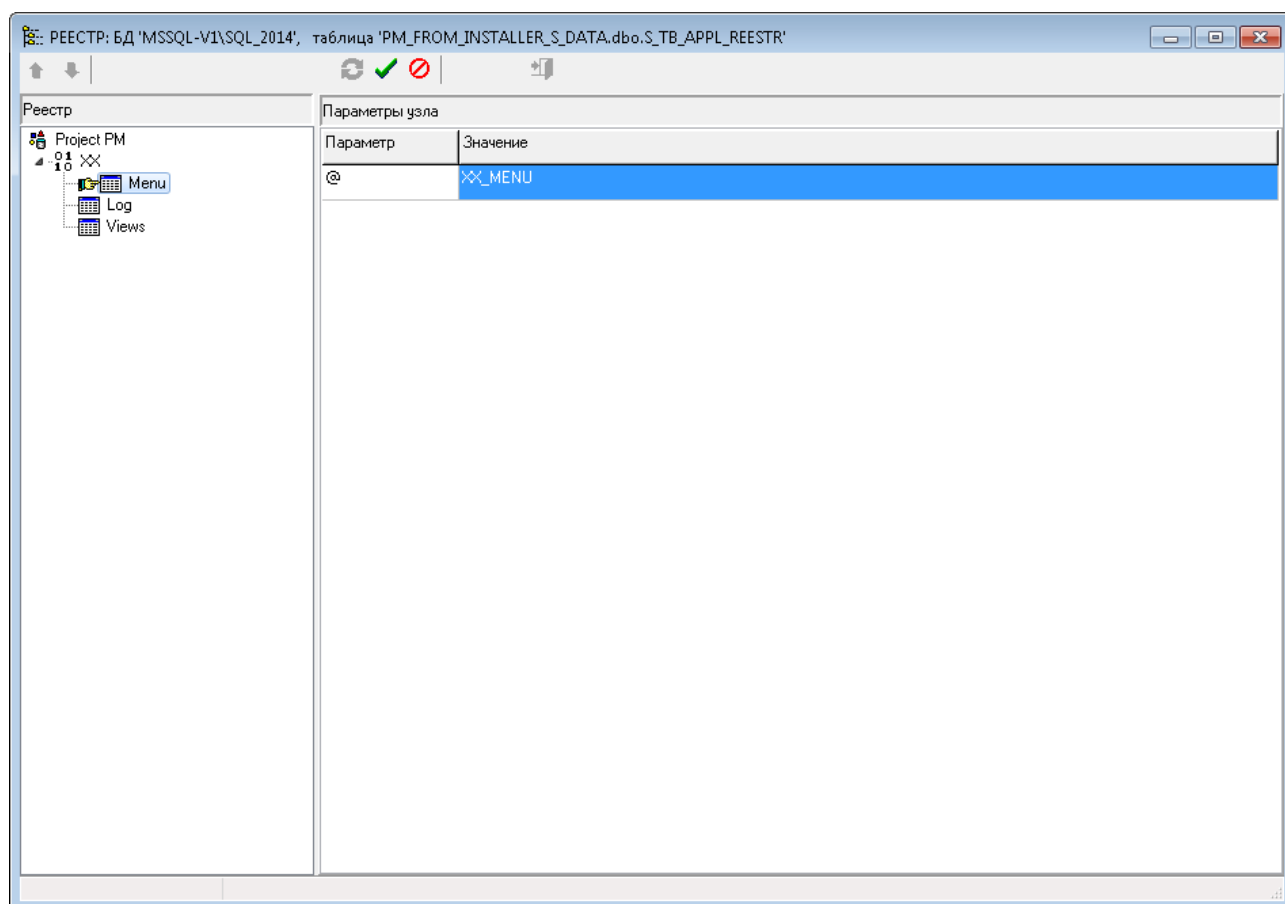
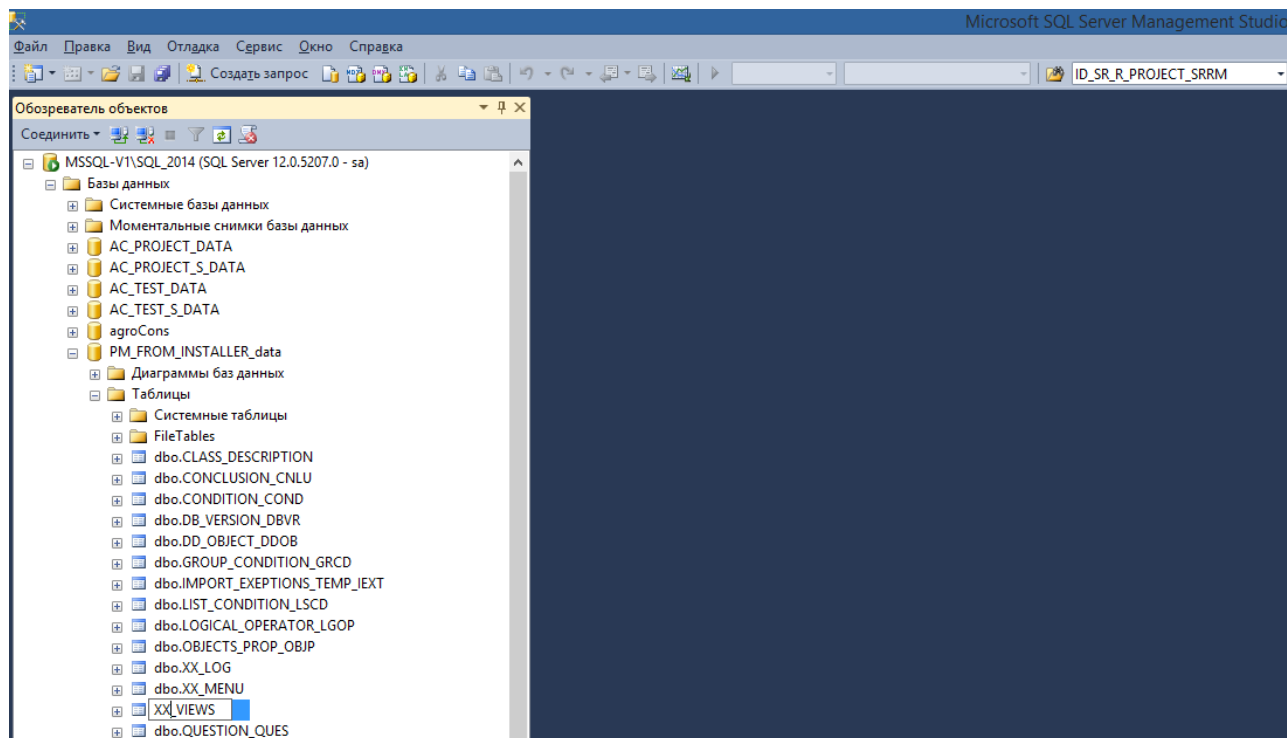
Чтобы создать проект с именем сборки, отличным от «**PM**», нужно действовать так же, как описано в главе [Шаг 1. Создание нового проекта](#). Т.е. создать (если нет) БД с необходимыми таблицами. Особенность заключается в дополнительном этапе после создания проекта – переименовании «Приложения» и, возможно, (не обязательно) «Проекта» в «Реестре приложений». Также желательно переименование служебных таблиц PM_MENU, PM_LOG и PM_VIEWS в «Реестре приложений» (Третий уровень) и в БД.

После создания из VS-проекта с помощью расширения «IMProjectWizard» получается «Реестр приложений», как на скриншоте выше. Если нужно назвать сборку иначе, например, XX_15_Z4.exe, то следует переименовать узел второго уровня «Приложения» из «PM» в «XX» и сохранить изменения.



После этого можно переименовывать сборку в XX_15_Z4.exe, ошибки не будет, все будет работать корректно.

Но тут нужно помнить, что служебные таблицы остались с именами PM_MENU, PM_LOG и PM_VIEWS. На работу приложения это не повлияет и ошибок не вызовет, но хорошим тоном в такой ситуации будет переименование этих таблиц в БД, и изменение их описания в Реестре «Описание служебных таблиц Приложения». Рекомендуются следующие названия: префикс <2ПервыхСимволаИмениПриложения> далее _MENU, _LOG и _VIEWS. Т.е. получится XX_MENU, XX_LOG и XX_VIEWS. Разумеется, переименование таблиц в БД и изменение их описания в Реестре надо выполнить одновременно.



Пользователь проекта. Система привилегий

До сих пор, работая с нашим проектом, мы использовали одного пользователя – s_dba. В рамках *S_Технологии*, это специальный пользователь, **Администратор проекта/БД**. На сервере БД создается “Имя входа” (MSSQL) или “Пользователь/Схема” (Oracle) с таким именем. Этому пользователю доступны все возможности проекта.

Обычно для прикладного пользователя все возможности бывают излишни и требуется ограниченные полномочий. Для реализации этих ограничений в *S_Технологии* имеется механизм **Пользователей проекта/Привилегий**. В отличие от пользователя s_dba, для **Пользователя проекта** не создается отдельного “Имени входа” (MSSQL) или “Пользователя/Схемы” (Oracle). Соединение с БД выполняется через специальное “Имя входа” – “Прокси пользователь”, которое мы создали на этапе “Шаг 1. Создание нового проекта”.

Пользователи проекта и Привилегии создаются и редактируются утилитой **pjmgr.exe**, подробная инструкция в документе “ИМ Руководство администратора системы.pdf”.

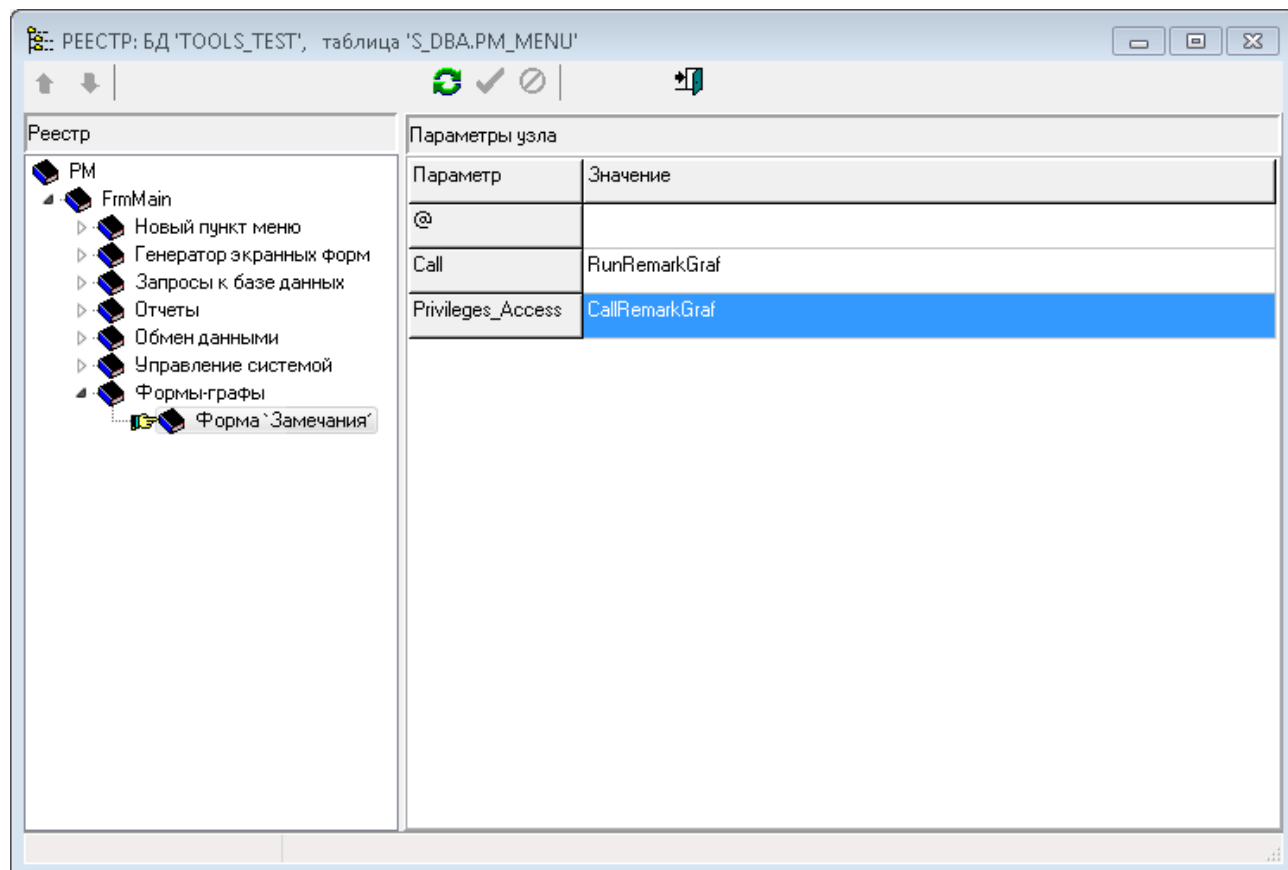
В Приложении 3 приведен состав и назначение привилегий на операции, которые создаются конфигуратором во время создания нового проекта.

Рассмотрим способы проверки **Привилегий** для **Пользователей проекта**.

Проверка привилегий в Меню проекта.

Есть два способа проверки **Привилегии** в Меню проекта.

Первый – явное задание **Привилегии** специальным параметром для пункта меню. На “Шаг 5. Подключение формы-графа к проекту”, в разделе “Сценарий 2. Вызов формы-графа” мы добавляли в Меню подузел – «Форма ‘Замечания’» и задавали ему «Параметр»-«Call» имя VS-метода. Для проверки **Привилегии** нужно добавить «Параметр»-«Privileges_Access», имя **Привилегии**, например “CallRemarkGraf”. В результате пункт меню будет иметь вид:



В результате, при старте приложения, S_Технология проверит имеется ли **Привилегия** с именем “CallRemarkGraf” и выдана ли она **Пользователю**, запускающему проект. Если эти условия не выполняются, данный пункт Меню не сформируется.

Второй способ проверки **Привилегии** в пункте Меню – неявный. Он применяется для “Справочников”. В разделе “Шаг 3. Использование таблицы в качестве справочника” мы добавляли в Меню вызов Справочника «Физическое лицо» ключевой фразой – “Class=SrPhysPersonSrppData”. Явного задания проверки **Привилегий** в пункте Меню нет, но тем не менее оно выполняется.

У класса-сущности «Физическое лицо» (*SrPhysPersonSrppEntity*) имеется описание свойств:

```
, PrivilegeRead = @"Default"
, PrivilegeWrite = @"Default"
```

Это означает, что для доступа к таблице-справочнику «Физическое лицо» (SR_PHYS_PERSON_SRPP) используются две привилегии со стандартными именами:

Привилегия на просмотр - R_<имя таблицы>

Привилегия на редактирование - W_<имя таблицы>

Т.е. R_SR_PHYS_PERSON_SRPP и W_SR_PHYS_PERSON_SRPP.

Соответственно, если не будет или не будет выдана **Пользователю Привилегия** R_SR_PHYS_PERSON_SRPP, пункт меню «Физическое лицо» вообще не сформируется, а если не будет **Привилегии** W_SR_PHYS_PERSON_SRPP, **Пользователь** не сможет редактировать справочник «Физическое лицо».

При использовании Сущности не как Справочник, а в составе формы-графа, свойства **PrivilegeRead/PrivilegeWrite** обычно удаляются из описания и используются другой набор **Привилегий**, см. ниже.

Проверка привилегий в Графе.

В разделе “Шаг 6. Редактирование графа”, “1.2. Свойства узла” уже упоминался набор **Привилегий**, проверяемый в форме-графе. При генерации Графа, для каждого Узла генерируются проверяемые **Привилегии**. По сравнению со Справочником, имеющим только две **Привилегии** – Просмотр/Редактирование, набор формы-графа значительно расширен. При этом имена Привилегий формируются по правилам:

Если у Сущности в Узле указаны Свойства “**PrivilegeRead = @"Default"**” и “**PrivilegeWrite = @"Default"**” будут взяты имена стандартных Привилегий – R_<имя таблицы>, W_<имя таблицы>:

PrivilegViewName	= 'R_<имя таблицы>'	- Привилегия на Просмотр таблицы
PrivilegEditName	= 'W_<имя таблицы>'	- Привилегия на Редактирование
PrivilegInsertName	= 'W_<имя таблицы>'	- Привилегия на Добавление записи
PrivilegDeleteName	= 'W_<имя таблицы>'	- Привилегия на Удаление записи
PrivilegCopyName	= 'W_<имя таблицы>'	- Привилегия на Копирование записи

Если Свойства **PrivilegeRead/PrivilegeWrite** не указаны, имена будут формироваться как служебный префикс **Привилегии**, а в конце – _<суффикс>, последние четыре символа из имени таблицы:

PrivilegViewName	= 'VIEW_<суффикс>'	- Привилегия на Просмотр таблицы
PrivilegEditName	= 'CHAN_<суффикс>'	- Привилегия на Редактирование
PrivilegInsertName	= 'BUILD_<суффикс>'	- Привилегия на Добавление записи
PrivilegDeleteName	= 'DEL_<суффикс>'	- Привилегия на Удаление записи
PrivilegCopyName	= 'COPY_<суффикс>'	- Привилегия на Копирование записи

Все имена **Привилегий** в схеме Графа можно изменить или вообще убрать их описание из схемы, что отключит проверку (не рекомендуется). **Привилегии** должны быть добавлены в БД и выданы нужным **Пользователям**.

В результате, при работе, Граф автоматически будет учитывать наличие соответствующих **Привилегий** и разрешать/запрещать **Пользователю** действия.

Проверка привилегий в коде приложения.

Помимо автоматической проверки **Привилегий**, описанной выше, иногда требуется проверка привилегии на какое-то отдельное нестандартное действие. Чаще всего это будет какая-то операция по “нестандартной” кнопке в форме Графа.

В разделе “Шаг 6. Редактирование графа. Добавление пользовательской формы и кнопки на ней” мы рассматривали, как добавить “нестандартную” кнопку в форме Графа. Рассмотрим, как ей можно управлять с помощью **Привилегии**.

Прежде всего проверка привилегия выдана/не выдана выполняется очень просто, с помощью метода класса **s_UserEntity**:

```
bool isPriv01 = s_UserEntity.IsPrivGranted("<имя привилегии>");
```

Используя метод **IsPrivGranted** можно проверить наличие **Привилегии** в любом месте приложения. Более подробно о классе **s_UserEntity** см. в документации **s_Vs.chm**.

Далее, у формы-графа есть виртуальный **FormOfNode SetButtonsEnability**. Перекроем его и дополним обработкой доступности нашей кнопки **myButton1** из примера в разделе “Шаг 6.”:

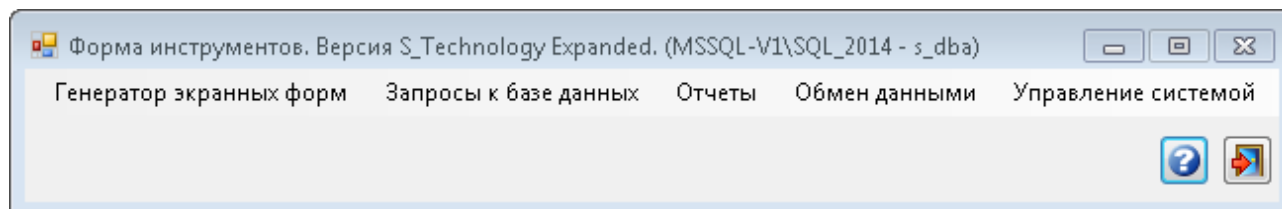
```
public override void SetButtonsEnability()
{
    base.SetButtonsEnability();
    myButton1.Enabled = s_UserEntity.IsPrivGranted("CallFormHello");
}
```

При наличии **Привилегии** "CallFormHello" кнопка будет доступна.

Более подробно про пользователей проекта и привилегии см. документ “ИМ Руководство администратора системы.pdf”.

Развитие приложения

Расширение Microsoft Visual Studio «IMProjectWizard» всегда создает один и тот же проект, типа PM_MSSQL.csproj. Т.е. это будет небольшая главная форма, снабженная меню с определенным набором пунктов и двумя кнопками.



Возможно, некоторые из пунктов меню не потребуются для рабочего проекта. Возможно, не потребуется меню вообще, а реализация вызовов будет выполнена в виде ленты или кнопок.

Как корректировать главное меню, было рассмотрено на Шагах 4 и 6 – использование утилиты **pjmgr.exe**. Был описан процесс добавления новых пунктов меню. Удаление пунктов меню делается еще проще, через то же контекстное меню.

Если меню главной формы не нужно вообще, можно просто удалить компонент меню из класса главной формы и все ссылки на него в VS-проекте. Класс формы – **FrmMain**, компонент **FormMainMenu**.

Допустима настройка и размеров главной формы. По умолчанию главная форма небольшая, фиксированного размера. При необходимости можно изменить размер или сделать форму Sizable. Можно добавить на форму новые компоненты.

Общий сценарий использования проекта

Рекомендуется следующий сценарий использования проекта.

1. Создать новый проект *S_Технологии* с использованием расширения VS «IMPProjectWizard».
2. Конфигурировать БД. Возможно, в процессе не будет использован Конфигуратор БД, тогда нужно его просто закрыть (не рекомендуется, т.к. даже если БД уже создана, конфигуратор выполнит проверку корректности).
3. Переименовать имя сборки VS-проекта и приложения в Реестре (если необходимо).
4. Изменить исходя из специфики проекта главное меню, вид главной формы и пр.
5. Наполнить проект модулями таблиц (сущностями), формами и т.д.
6. Добавить дополнительные формы, классы и пр., которые рекомендуется оформлять как потомки классов *S_Технологии* (не обязательно).

Приложение 1. Примеры применения Генератора форм

Формы, графы

Как перекрыть обработчик базовой кнопки формы узла (показ графа-карточки)

Для показа карточки нужно в форме (потомок *FormOfNode*) перекрыть обработчик показа карточки

```
protected override void AcCard()
{
    s_NamedObjects no = new s_NamedObjects();
    no[Const.par_node_hide_list_ + "0"] = true; // скрываем сетку
    no[Const.par_node_value + "0"] = CurrentID; // задаём значение узла
    no[Const.par_node_condition + "0"] =
string.Format("{0}={1}", this.KeyField, this.CurrentID); // ограничиваем выборку 1 строкой
    no[Const.par_hand_card] = true; // признак того, что карточка создана
    no[Const.par_task_caption] = "ФИО"; // заголовок карточки

    WFGraf.ShowGraf(new PpPhysFioSchema.GetTest(), no);
}
```

Для многих действий существуют виртуальные методы: добавить-AcAdd, удалить-AcDelete;

Но часть кнопок т.к. Сохранить и Откатить являются общими для всех форм графа и принадлежать не конкретной форме узла, а базовой форме графа.

Как перекрыть обработчик базовой кнопки формы графа

Во-первых, нужно создать потомка *FormOfGraf* и в нужном графе указать, что для главной формы графа используется не базовая форма.

```
"    main_form_name                                ='MyProject.FrmCustomGrafForm, MyProject
'" +
```

Во-вторых, в форме *FrmCustoGrafForm* перекрыть нужный обработчик

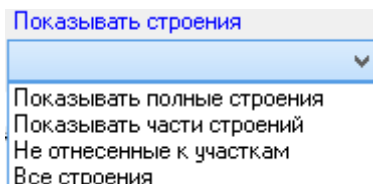
```
protected override void AcCommit()
{
    // так можно получить по имени сущности (если такая сущность одна)
    WFNode pp = Graf.NodeByEntityName("MyEntity") as WFNode;
    // так можно получить форму, которая «лежит» под узлом
    // и, например, вызвать какую то проверку
    if ((pp.Form as MyForm).Check ())
    {
        base.AcCommit();
    }
}
```

```
    }
}
```

Таким образом мы сможем выполнить какую-нибудь проверку в обработчике нажатия кнопки «Сохранить»

Как создать фильтр со списком значений статичный (custom)

Рассмотрим пример, в котором в каком-то взятом графе необходимо добавить комбобокс со значениями:



Для этого нужно создать узел по примеру ниже:

```
"
<Node" +
"    Caption          ='Показывать строения'" +
"    ind              ='22'" +
"    Node_Name        ='node_Pokaz'" +
"    Name_Field_rus    ='Показатель'" +
"    ID_Field          ='ID_Field'" +
"    Name_Field        ='Name_Field'" +
"    enabled_node_statuses      ='ns_master_filter' " +
"    enabled_filter_types      ='ft_combobox' " +
"    required              ='False'" +
"    >" +
"    <CustomTableAttributes>" +
"        <Attribute Name  ='ID_Field'" +
"            RusName      ='Код'" +
"            ReadOnly      ='true'" +
"            default_width ='30'" +
"        /> " +
"        <Attribute Name  ='Name_Field'" +
"            RusName      ='Название'" +
"            ReadOnly      ='true'" +
"            default_width ='400'" +
"        /> " +
"    </CustomTableAttributes>" +
"    <CustomTableRows>" +
"        <Row" +
"            cell_0      ='INCLUD_BLBV is null'" + //условие, накладываемое на подчиненную
сетку
"            cell_1      ='Показывать полные строения'" +
"        /> " +
"        <Row" +
"            cell_0      ='not INCLUD_BLBV is null'" +
"            cell_1      ='Показывать части строений'" +
"        /> " +
"        <Row" +
"            cell_0      ='ID_BLBV is null'" +
"            cell_1      ='Не отнесенные к участкам'" +
"        /> " +
"        <Row" +
"            cell_0      ='1=1'" +
"            cell_1      ='Все строения'" +
"        /> " +
"    </CustomTableRows>" +
"    > " +
" </Node> " +
```

Отдельно следует указать, что значение узла в связь переноситься без дополнительных обработок. Таким образом можно использовать не только константы в значениях, но и sql которые будут дополнять выражение связи с узлом. Подробнее о параметра, представленных в примере см. в описании схемы графа

Как создать фильтр со списком значений заполняемых в runtime

Во-первых, нужно создать узел с CustomAttributes

```
" <Node" +
"   Caption      ='Период'" +
"   ind          ='5'" +
"   Name_Field_rus  ='Наименование'" +
"   Name_Field     ='Name_Field'" +
"   ID_Field       ='ID_Field'" +
"   Enabled_node_statuses='ns_show_data;ns_comboBox' " +
"   Default_node_status  ='ns_show_data'" +
"   MultiSelect     ='False' " +
"   privileg_view_name='VIEW_ALWAYS'" +
"   row_num         ='2'" +
"   col_num         ='3'" +
" >" +
"   <CustomTableAttributes>" +
"       <Attribute Name  ='ID_Field'" +
"           RusName      ='Код'" +
"           ReadOnly     ='true'" +
"           default_width='45'" +
"       /> " +
"       <Attribute Name  ='Name_Field'" +
"           RusName      ='Название'" +
"           ReadOnly     ='true'" +
"           default_width='400'" +
"       /> " +
"   </CustomTableAttributes>" +
" </Node>" +
```

Во-вторых, подписаться на события

```
graf.CreateAdapterTableEvent += self_graf_custom_Create_adapter_Table;
graf.ReloadAdapterTableEvent += self_graf_custom_Reload_adapter_Table;
```

Объект графа можно получить, например, в форме узла.. В обработчике события **CreateAdapterTableEvent** мы должны для узла 5 задать таблицу. В данном случае это будет таблица с двумя полями ID_Field,Name_Field типа string. Код обработчика представлен ниже:

```
/// <summary>
/// создание новых таблиц для дата-адаптера
/// </summary>
/// <param name="e"></param>
void self_graf_custom_Create_adapter_Table(object sender, CustomDataTableEventArgs
e)
{
    e.Result = false;
    SNode curr_node = graf.Node_(e.NodeUID);
    if (curr_node == null) return;

    if (Func.SameStr(graf.GetNodeUID(curr_node), "5"))
    {
        e.Table = new DataTable();

        e.Table.Columns.Add(Const.def_ID_Field, typeof(String));
        e.Table.Columns.Add(Const.def_Name_Field, typeof(String));
    }
}
```

```

        e.Result = true;
    }
}

```

В обработчике события **ReloadAdapterTableEvent** происходит заполнение таблицы нашего динамического узла.

```

/// <summary>
/// перезагрузка содержимого таблиц
/// </summary>
/// <param name="e"></param>
void self_graf_custom_Reload_adapter_Table(object sender, CustomDataTableEventArgs
e)
{
    SNode currNode = graf.Node_(e.NodeUID);
    if (currNode == null || e.Data == null) return;

    e.Table.Rows.Clear();

    (e.Data as s_DataObject).Rows.Clear();

    // период
    if (Func.SameStr(graf.GetNodeUID(currNode), "5"))
    {
        // код заполнения таблицы e.Table
    }
}

```

Как подписаться на обработку всех графов

Для того, что бы сделать какую то общую обработку графов или что бы создать класс-адаптер к графу нужно подписаться на prepare графов.

```
SGraf.OuterEventsObj.OuterPrepareGrafEvent += outer_prepare_grafs;
```

В методе outer_prepare_grafs

```

public void outer_prepare_grafs(CustomGrafEventArgs e)
{
    SGraf lgraf = e.Graf as SGraf;
}

```

lgraf – полнофункциональный граф, у которого можно подписывать на изменение значений узлов, получить доступ к узлу по индексу, менять параметры узла, работать с дата-классом узла.

Для того, чтобы не было утечек памяти нужно отписываться от событий.

Как делать ручную карточку (привязка своих контролов к атрибутам выборки)

Карточка, которая доступна по умолчанию в форме FormOfNode не всегда может соответствовать требованиям. Например, нужно сделать группы полей, определить свой порядок следования атрибутов дата-класса. Для этого, нужно в наследнике FormOfNode (который подключён к графу) перейти на TabPage с именем **tabPage_card** и разместить необходимые настройки. Далее нужно перекрыть виртуальный метод SetupView у формы

```

protected override void SetupView(s_View p_View)
{
    if (p_View == null) return;


    base.SetupView(p_View);

    p_View.Options = p_View.Options
        | s_ViewOption.ConfSave
        | s_ViewOption.ImportFile
        ;
    p_View.Attributes[ColumnIdSOkatoName].ControlView = textBox1;
}

```



```
p_View.Attributes[ColumnIdSOkatoName].ControlCaption = label1;
```

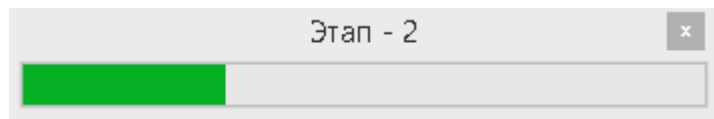
При этом в textbox1 будет записываться значение из атрибута с именем ColumnIdSOkatoName, а title этого атрибута будет записываться в контрол с именем label1. При этом, если вы переопределяете ControlView для ссылочного атрибута, рядом с textbox будет помещена кнопка выбора  это нужно учесть при проектировании интерфейса карточки

Как подписаться на изменение узла графа

Для того, чтобы подписаться на изменение узла графа нужно воспользоваться событием **NodeValueChangedEvent** у класса BGraf или его наследников.

Описание класса **NodeValueChangedEvent** смотрите в документации

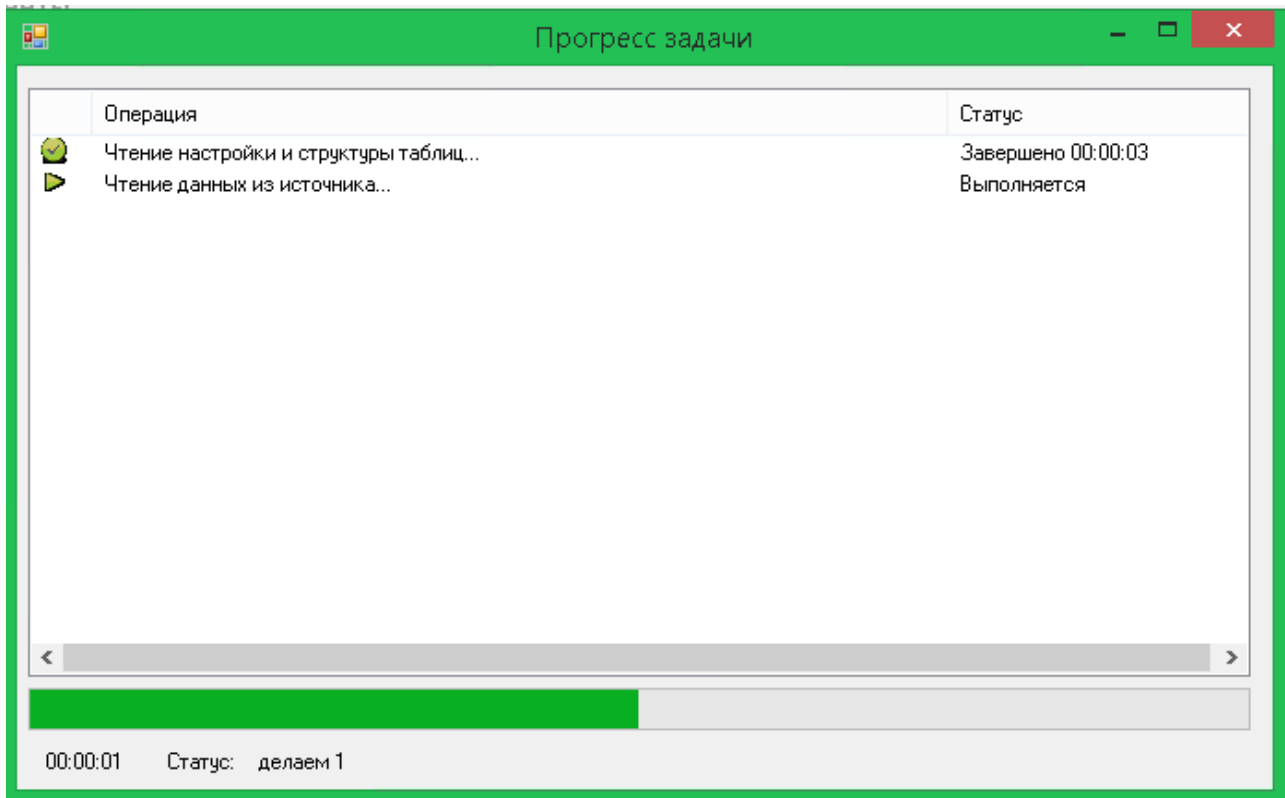
Как использовать форму с прогрессом (простая)



Код примера:

```
using (FormProgress fGaugeForm = new FormProgress())
{
    fGaugeForm.StartProgress(s_Application.GetText("fm_Res", "threatened_from") + 10, 10);
    try
    {
        for (int i = 0; i < 10; i++)
        {
            fGaugeForm.MoveProgress("Этап - " + i);
            System.Threading.Thread.Sleep(1000);
        }
    }
    finally
    {
        fGaugeForm.EndProgress();
    }
}
```

Как использовать форму с прогрессом (с этапами, как в импорте)



Код вызова данной формы(расположена в InuETL.FORMS):

```
fm_InuImportProgress fmProgress = fm_InuImportProgress.Run();
fmProgress.Text = "Прогресс задачи";
    // первый этап
    fmProgress.startStaging("Чтение настройки и структуры таблиц...", 3);
    // делаем 1
    Thread.Sleep(1000);
    fmProgress.progressStep();
    // делаем 2
    Thread.Sleep(1000);
    fmProgress.progressStep();
    // делаем 3
    Thread.Sleep(1000);
    fmProgress.progressStep();
    fmProgress.finishStaging();
    // второй этап
    fmProgress.startStaging("Чтение данных из источника...", 2);
    Thread.Sleep(1000);
    fmProgress.progressStep("делаем 1");
    Thread.Sleep(1000);
    fmProgress.progressStep("делаем 2");
    fmProgress.finishStaging();
    fmProgress.FormClose();
```

Как расположить фильтры в несколько рядов (задание каждому отбору своего места)

Для того, чтобы расположить фильтры в несколько рядов в строго заданном порядке нужно сделать следующее.

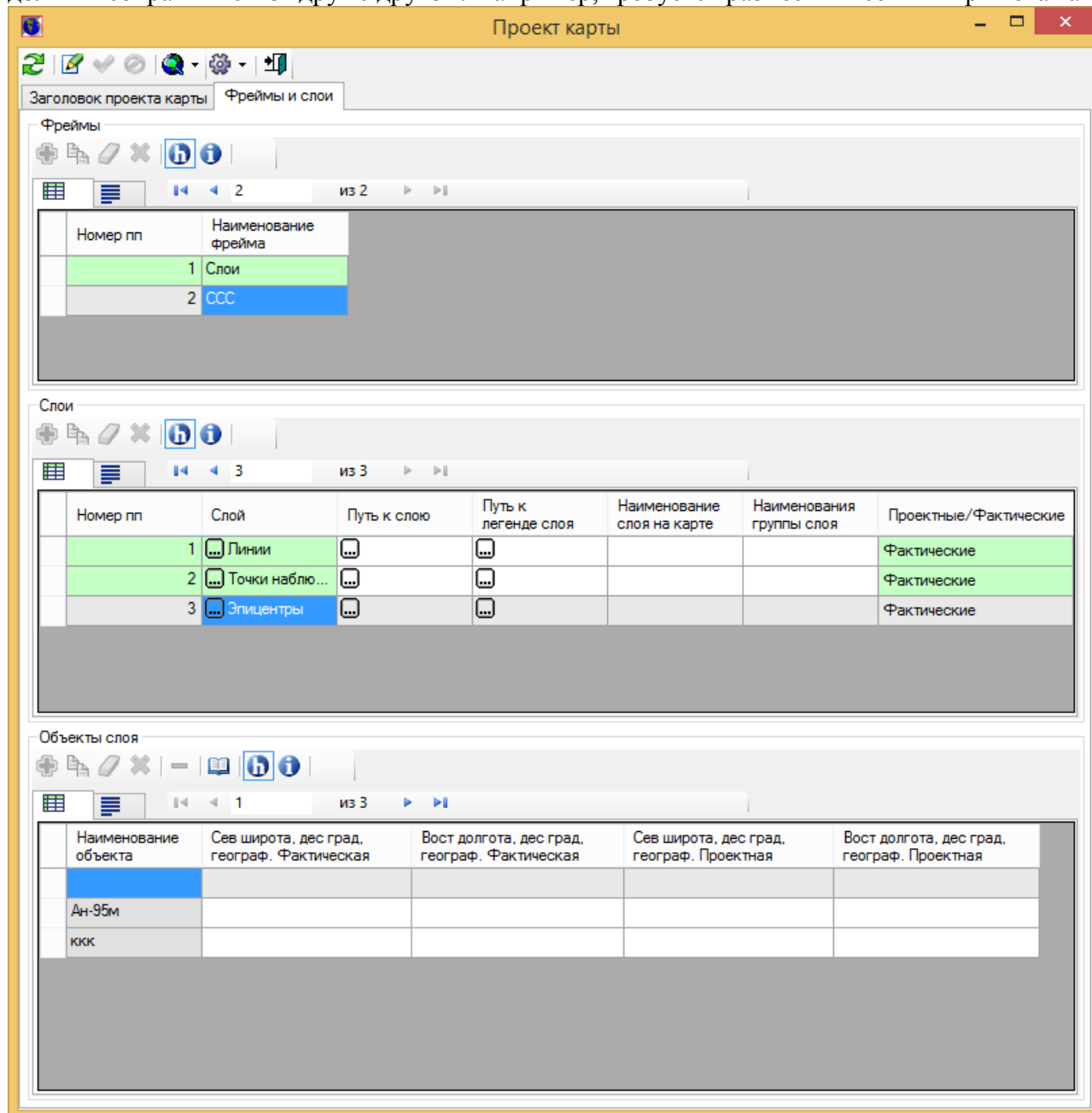
Во-первый, установить свойство графа (в схеме графа) *auto_position = 'false'*

Во-вторых, для каждого фильтра выставить значения строки и столбца в котором должен располагается фильтр.

```
"      row_num      = '1' " +
"      col_num      = '1' " +
```

Как использовать базовые панели формы для размещения своих элементов (мастер->деталь->деталь)

Иногда требуется разместить узлы графа нестандартно, но, в то же время, узлы должны сохранять связь друг с другом. Например, требуется разместить сетки в три «этажа».



```

graf = new WFGraf(false);
fm_MmPrjmapFullCardMR self_form = new fm_MmPrjmapFullCardMR(graf);
graf.LoadFromXML(«текст схемы графа»);

graf.ParentControls["0"] = self_form.tabPage_Card;
graf.ParentControls["1"] = self_form.gb_top_master_panel;
graf.ParentControls["2"] = self_form.gb_midle_panel;
graf.ParentControls["3"] = self_form.gb_low_panel;
graf.SelfsupportingNodeStatuses = false;

graf.Node_("0").DefaultStatus = NodeStatus.ns_master_form;

```

```

graf.Node_("1").DefaultStatus = NodeStatus.ns_master_form;
graf.Node_("2").DefaultStatus = NodeStatus.ns_detail_form;
graf.Node_("3").DefaultStatus = NodeStatus.ns_detail_form;

if (!graf.Prepare()) return;

self_form.ShowDialog();
graf.Dispose();

```

Где `public partial class fm_MmPrjmapFullCardMR : FormOfGrafHandCrafted` форма основание для визуальных компонентов графа унаследована от формы `FormOfGrafHandCrafted`.

Список `graf.ParentControls` содержит пары «UID узла – контрол». В приведенном выше примере узел с `UID = "0"` помещается на первую закладку табконтрола, а три остальных помещаются на три панели второй закладки.

Как задать доп. условие на узел

Дополнительное условие на выборку узла можно наложить с помощью свойства `condition_before_open` в значении этого свойства нужно поместить sql условие, которое будет добавляться во фразу `where`. Например, данное условие выберет данные только с `ID_VID=1`.

```
"      condition_before_open = '(ID_VID=1) ' " +
```

Как по кнопке сменить выборку (смена selectionName)

Рассмотрим пример перехода к от одной выборки к другой. Место действия – наследник формы `FormOfNode`. На

```

/// <summary>
/// Имя параметра признака расширенного вида
/// </summary>
string _showExtFields = "_showExtFields";

/// <summary>
/// Признак расширенного вида
/// </summary>
private bool IsExtend
{
    get
    {
        if (!Node.NodeProp.ContainsKey(_showExtFields))
            Node.NodeProp[_showExtFields] = false;

        return (bool)Node.NodeProp[_showExtFields];
    }
    set
    {
        Node.NodeProp[_showExtFields] = value;
    }
}

/// <summary>
/// Обработчик кнопки перехода в расширенных вид (есть в FormOfNode)
/// </summary>
protected override void AcExtend()
{
    IsExtend = !IsExtend;
    if (Graf != null) Graf.NodeValueChangedEvent -= self_graf_onNodeValueChanged;
    base.AcExtend();
}

/// <summary>

```

```

/// Переключаем метод создания сущности
/// </summary>
/// <returns></returns>
public override s_EntityObject CreateEntity()
{
    PpPhysPersEntity phpr = new PpPhysPersEntity();

    if (IsExtend)
        phpr.FullExtend(s_EntityAction.Prepare); // готовим сущность к показу в
расширенном виде
    else phpr.fast_PHPR(s_EntityAction.Prepare); // готовим сущность к показу в
стандартном виде

    return phpr;
}

```

Как добавить кнопку на базовую панель формы (через функции)

Для того, чтобы добавить кнопку runtime на панель базовой формы можно воспользоваться функцией (последним элементом перед кнопкой `tb_fiscal`) `tbShowCardShash = bd_Navigator.AddButtonOnToolStrip("Открыть список шашек", Properties.Resources.ShowProb, ShowCardShash);` где:

- `bd_Navigator` - панель на которую нужно добавить кнопку
- `AddButtonOnToolStrip` расширение (функция для добавления кнопки `using IM.Functions`);
- "Открыть список шашек" – tooltip кнопки;
- `ShowProb` – картинка, которую нужно поместить на кнопку;
- `ShowCardShash` – имя метода, который нужно вызывать при нажатии на кнопку.

Как управлять доступностью кнопок в форме узла

Для управления доступностью кнопок нужно перекрыть виртуальный метод формы `FormOfNode` **SetButtonsEnability**. Например, нужно сделать доступной кнопку `tbShowCardShash` только в режиме просмотра. Код будет выглядеть так:

```

public override void SetButtonsEnability()
{
    base.SetButtonsEnability();
    tbShowCardShash.Enabled = this.s_Status == s_FormStatus.Browse;
}

```

Как назначить картинку для колонки и обработать клик по ней

Допустим нужно сделать выбор данных или запуск произвольной формы по ссылке псевдо-ссылочной кнопке (у атрибута, который не является ссылкой).

```

protected override void SetupView(s_View p_View)
{
    if (p_View == null) return;
    base.SetupView(p_View);
    p_View.Options = p_View.Options
        | s_ViewOption.ConfSave
        | s_ViewOption.ImportFile
        ;
    // во-первых, назначить картинку для столбца
    p_View.Attributes[ColumnLogin].Image =
s_Application.Image16List.Images["s_Definit2"]; ;
    // во-вторых, подписаться на событие CellImageClick
    p_View.DataView.CellImageClick += new
s_ViewCellImageClickEventHandler(p_View_CellImageClick);
}

```

В обработчике клика, проверить что клик идёт по нашей колонке

```
void p_View_CellImageClick(object sender, s_ViewCellImageClickEventArgs e)
{
    if (e.ViewAttribute.Name == ColumnLogin)
    {
        // делаем что-то
    }
}
```

Сущности

Как добавить бизнес-логику в сущность

Бизнес-логика (правила) требуются для проверки данных на соответствие определённым правилам или шаблонам. Данную проверку нужно делать в заготовленном методе. Метод носит название **BusinessRulesCheck**. Он расположен в базовом предке сущностей GrafEntity

В качестве примера, код проверки типа документа:

```
public override bool BusinessRulesCheck(ref string ErrorText,
s_EntityInstanceValidatingEventArgs e)
{
    if (e.Row[ColumnIdRTypeDoc].ObjToID() != 4)
    {
        // устанавливаем сообщение, которое нужно показать на экране
        ErrorText = "Документ должен быть типа 4";
        // указываем, что проверка бизнес правил была не успешной
        return false;
    }
    return true;
}
```

Как задать сортировку дата-класса

Есть три способа задать сортировку для дата-класса:

1. Свойство **Data.Sort** возвращает или задает атрибут или атрибуты для сортировки, а затем — порядок сортировки для представления выборки. Для доступа к отсортированным строкам следует использовать свойство DataViewRows. Например,

Sort= "myFirstSortColumn, mySecondSortColumn desc"

Примечание: Свойство **Sort** не может использоваться в режиме пейджинга

2. Свойство дата-класса **OrderByClause**. Список OrderByClause формируется программистом в методе GetSQL и не должен меняться в дальнейшем. Значения в данном списке имеют sql синтаксис фразы order by.
3. Свойство **UserOrderByClause** определяет сортировку запроса выборки. Может использоваться программистом в runtime. Смена значения данного свойства повлияет на сортировку только после переоткрытия выборки. UserOrderByClause имеет более высокий приоритет перед OrderByClause

Как назначить для сущности показ графа (перекрытие метода ShowData)

Метод **ShowData** используется классами s_технологии в следующих ситуациях:

- класс s_View использует реализацию при отображении ссылок Lookup-колонок;

- метод `s_View.RefExec` при отображении выборки дата-класса, заданного именем.

(подробнее в `s_VS.chm`)

Пример перекрытия `ShowData`:

```
public override void ShowData(object parameters)
{
    s_EntityShowingEventArgs __e = new s_EntityShowingEventArgs();
    OnShowing(__e);
    if (__e.Cancel) return;

    s_NamedObjects lParams = GetShowDataParams(parameters);
    ShowGraf("0", new PpPhysAddrSchema().GetText(), lParams);
    OnShowned(EventArgs.Empty);
}
```

Данный код при запросе показа сущности запустит граф `PpPhysAddrSchema` у которого главным узлом будет узел с `UID=0` и будут переданы параметры `lParams`.

Как передать доп. условие или параметры в вызываемую по ссылочной кнопке сущность

Для передачи параметров в вызываемую сущность нужно перекрыть у дата-класса метод **OnLookupShowing**

и использовать свойство **ShowDataParams** сущности(сущность должна быть унаследована от `GrafEntity`).

Например,

```
protected override void OnLookupShowing(s_LookupShowingEventArgs e)
{
    base.OnLookupShowing(e);
    GrafEntity ent = e.RefData.Entity as GrafEntity;
    if (ent != null)
    {
        ent.ShowDataParams[Const.par_node_value + "df1"] = 29;
    }
}
```

Данный код передаёт некий параметр `par_node_value` в вызываемую сущность. Обработка таких параметров может быть в нескольких местах. В самой `ShowData` (вызов `showData` происходит всякий раз, когда заказан переход по ссылочной кнопке в сетке или карточке). В `prepare графа`, в случае, если у вызываемой сущности перекрыта `ShowData` и происходит не показа справочника (простой сущности), а заказывается показ какого-то графа. И тогда, данный пример приведёт к установке значения узла `df1` в 29.

Так же можно обработать `ShowDataParams` если перекрыть у вызываемой сущности метод `ShowData`.

Как определить набор полей для выборки

Есть три способа для определения набора полей, которые должны выводиться в сетку. Первый способ, в методе `GetSQL` дата-класса поставить условие на имя выборки и, в зависимости от имени, выдавать разный запрос (не выбирается атрибут – не показывается).

```
protected override void GetSQL()
{
    if (SelectionName.Equals("MyFirstSelectionName"))
    {
        SelectClause.Add("select " + ClassInfo.TableAlias + ".* ");
        FromClause.Add("from " + TableName + " " + ClassInfo.TableAlias);
    }
    else if (SelectionName.Equals("MySecondSelectionName"))
    {

```

```

        SelectClause.Add("select " + ClassInfo.TableAlias + ".* ");

        SelectClause.Add(", " +
"getFIO".ToFullObjectName(s_DatabaseObjectType.Function) + "(id) as FIO");
        FromClause.Add("from " + TableName + " " + ClassInfo.TableAlias);
    }
}

```

Это вариант работает только для простых атрибутов, ссылочные атрибуты формируются на другом уровне и для того, чтобы исключить ссылочный атрибут нужно воспользоваться параметром атрибута **UseIn** (второй способ регулирования набора полей) `UseIn='Default;FullList;ById;MyFirstSelectionName;MySecondSelectionName'`

И третий способ - это перекрыть виртуальный метод сущности **GetGridColumns**.

Например:

```

public override string GetGridColumns()
{
    string result = "";
    if (SelectionName == "MyFirstSelectionName")
    {
        result = ColumnStatus
            + ";" + ColumnFIO
            + ";" + ColumnPasport
            + ";" + ColumnDateBirth
            + ";" + ColumnYearBirth
            + ";" + ColumnGender
            + ";" + ColumnIdRNationalityName
            + ";" + ColumnAdress;
    }
    if (SelectionName == "MyFirstSelectionName")
    {
        result = ColumnDateDeath
            + ";" + ColumnComment
            + ";" + ColumnAeAdmUnitName
            + ";" + ColumnDateIn
            + ";" + ColumnDateOut;
    }
    return result;
}

```

Как создать вычисляемое поле

Для того, чтобы добавить вычисляемое поле в выборку нужно сделать пару шагов.

Во-первых, определить вычисляемое поле в виртуальном методе **DefineCalculate**

```

protected override void DefineCalculate(s_DataObject data)
{
    base.DefineCalculate(data);
    data.AddStringCalculate("MyCalculateFieldName", 98, "Имя вычисляемой колонки");
}

```

Таким образом можно добавить строковое вычисляемое поле `MyCalculateFieldName` длиной 98 и с заголовком «Имя вычисляемой колонки». Строка это универсальный тип колонки, но так же можно создать типизированную колонку, например через метод `AddDoubleCalculate` (более подробно смотрите в документации `s_Vs.chm`)

Во-вторых, нужно заполнить это поле. Делается это в методе **Calculate**:

```

protected override void Calculate(DataRow row)
{
    if (row == null)
        return;
    if ((row.RowState == DataRowState.Detached) || (row.RowState ==
DataRowState.Deleted))
        return;
}

```



```

        base.Calculate(row);
        row["MyCalculateFieldName"] = 1 + 2; // вычисляем значение для колонки
    }

```

Как обработать изменение атрибута

Иногда, требуется подписаться на изменение атрибута и выполнить действие. В этом нам поможет метод **OnAttributeChanged**. Например, сделаем лог изменений в консоль вывода

```

protected override void OnAttributeChanged(s_AttributeChangedEventArgs e)
{
    base.OnAttributeChanged(e);
    // пишем в консоль output изменение значения атрибута(ячейки)
    Console.Out.WriteLine(string.Format("Изменили {0} на значение {1}", e.Attribute.Name,
e.Attribute[e.Row]));
}

```

Как добавить ссылочное поле и наложить доп. условие на него

Основные понятия:

SourceClass (**TInuObjectWorkObwaData**) - исходный дата-класс

SourceSelection - выборка: **Default|For_View|ByID|ByDateFactBeginAndEnd**
при отображении строк которой надо показать
для каждой строки определение сотрудника, зависящее от некоей
даты в этой строке

(колонка **DATE_FACT_END_OBWA**)

RefClass (**EmployeeFioData**) - ссылочный класс. Таблица этого класса содержит
определение ФИО сотрудника. Для одного
и того же Id сотрудника в таблице может быть несколько таких
определений

с непересекающимися интервалами действия (действует с - по)

LookupColumn (**NAME_RESPONCE_EXECUTOR**) - ссылочная колонка в выборке
SourceClass.SourceSelection, ссылающаяся на RefClass

RefSelection (**FioByDate**) - ссылочная выборка в классе **EmployeeFioData**,
соответствующая по lookup-колонке

NAME_RESPONCE_EXECUTOR.

s_dba.check_date - функция Oracle с тремя параметрами:

pDateIn - дата начала временного интервала. Может быть null

pDateOut - дата конца временного интервала. Может быть null

pDateActual - дата, проверяемая на принадлежность временному

интервалу,

определенному параметрами pDateIn, pDateOut. Не может

быть null

Реализация:

На уровне SourceClass (TInuObjectWorkObwaData)

1. LookupColumn должна быть включена в каждую из выборок

Default|For_View|ByID|ByDateFactBeginAndEnd

Определение LookupColumn выглядит так:

[IMDataClassColumnNameAttribute

```
(
    Name = "NAME_RESPONCE_EXECUTOR"
    , Type = s_AttributeType.Lookup
    , UseIn = @"Default;For_View;ByID;ByDateFactBeginAndEnd"
    , RefClassName = @"EmployeeFioData.ByDate"
    , KeyColumns = @"RESPONS_OBWA"
    , Title = @"Ответственный исполнитель"
)]
```

2. Метод **GetSQL**. Накладываем дополнительное условие. Во фразу WHERE текста формируемого запроса для SourceSelection **Default|For_View|ByID|ByDateFactBeginAndEnd**

добавить вызов функции **s_dba.check_date**, где:

pDateIn, pDateOut - колонки DATE_IN, DATE_OUT из таблицы RefClass

(**EmployeeFioData**)

pDateActual - колонка **DATE_FACT_END_OBWA** или значение на случай

DATE_FACT_END_OBWA is null

```
if (SelectionName.ToUpper().In("DEFAULT", "FOR_VIEW", "BYID",
"BYDATEFACTBEGINANDEND"))
```

```
{
```

```
    WhereClause.Add("s_dba.CHECK_DATE({0}, {1}, nvl({2}, :pActualDate))"
```

```
        .Format2(EmployeeFioEntity.ColumnDateIn,
```

```
EmployeeFioEntity.ColumnDateOut
```

```
        , TinuObjectWorkObwaEntity.ColumnDATE_FACT_END_OBWA));
```

```
    SetParameter("pActualDate", <?>); //используемая дата когда
```

```
DATE_FACT_END_OBWA is null
```

```
}
```

(В тексте запроса можно ссылаться на колонки из **EmployeeFioData**, поскольку соответствующая таблица будет автоматически присоединена

как ссылочная таблица для **NAME_RESPONCE_EXECUTOR**)

3. Метод **OnLookupShowing**. Для lookup-колонки **NAME_RESPONCE_EXECUTOR**

предусмотреть определение значения параметра-даты

для ссылочной выборки **EmployeeFioData.ByDate**(см. п.2 для RefClass)

```
protected override void OnLookupShowing(s_LookupShowingEventArgs e)
```

```
{
```

```
    base.OnLookupShowing(e);
```

```
    if (e.Cancel)
```

```
        return;
```

```
    if
```

```
(e.LookupName.SameText(TinuObjectWorkObwaEntity.ColumnNAME_RESPONCE_EXE
CUTOR))
```

```
{
```

```
    e.RefData.ParamByName("pActualDate").Value =
```

```
    !e.Row.IsNull(TinuObjectWorkObwaEntity.ColumnDATE_FACT_END_OBWA) ?
```

```
        e.Row[TinuObjectWorkObwaEntity.ColumnDATE_FACT_END_OBWA]
```

```
        :
```

```
        <?>; //используемая дата когда значение DATE_FACT_END_OBWA is null
```

```
;
```

```

    }
}

```

На уровне RefClass (EmployeeFioData)

1. На уровне класса определить

```

[IMDataClassAttribute(
    ...
    , LookupResultAgg = @"trim(@FAMILY_EMFI)||' '||trim(@NAME_EMFI)"
    ...
)]
[IMDataClassSelectionAttribute(
    Name = @"ByDate"
    , DefaultCaption = @"На дату"
    , LookupKeyColumns = EmployeeFioEntity.ColumnIdEmployee
    , IsList = true
)]

```

2. Метод **GetSQL**. Во фразу WHERE для выборки **ByDate** добавить вызов функции **s_dba.check_date**, где:

pDateIn, pDateOut - колонки **DATE_IN_EMFI**, **DATE_OUT_EMFI** из таблицы **EMPLOYEE_FIO_EMFI**
 pDateActual - определяется как параметр. Когда выборка **ByDate** открывается по lookup-кнопке,
 значение этого параметра будет сформировано программистом в методе OnLookupShowing
 класса **TInuObjectWorkObwaData** (см. п.3 выше)

```

if (SelectionName.SameText("ByDate"))
{
    WhereClause.Add("s_dba.CHECK_DATE({0}, {1}, :pActualDate)"
        .Format2(EmployeeFioEntity.ColumnDateIn, EmployeeFioEntity.ColumnDateOut));
}

```

Приложение 2. Состав и назначение таблиц базы данных в новом проекте

В процессе создания нового проекта конфигуратор проекта записывает в базу данных несколько таблиц, которые необходимы для работы инструментальных средств. Ниже приведено назначение и состав этих таблиц. В зависимости от функций проекта, некоторые из этих таблиц должны быть переданы конечному пользователю разработанной системы.

Таблица	Назначение таблицы	С какого комплекта входит в поставку	Тип таблицы	Данные в таблице
ДОП ДАННЫЕ ОБЪЕКТА (DD_OBJECT_DDOB)	Объекты Windows: Xls, doc, jpeg и т.д.	Standard	Оперативная таблица инструментов	Нет
ВЕРСИЯ БД (DB_VERSION_DBVR)	Версия основной базы проекта	Standard	Системный справочник пользователя-разработчика	Строка с версией БД 1
ОПИСАНИЕ КЛАССА (CLASS_DESCRIPTION)	Описания классов проекта	Standard	Системный справочник пользователя-разработчика инструментов. Содержит классы разработчика инструментов. Пополняется пользователем-разработчиком.	Строки описания классов инструментальных средств, входящих в текущий комплект поставки.
СВОЙСТВО КЛАССА (OBJECTS_PROP_OBJP)	Свойства классов проекта	Standard	Системный справочник пользователя-разработчика инструментов. Содержит классы разработчика инструментов. Пополняется пользователем-разработчиком.	Строки свойств классов инструментальных средств, входящих в текущий комплект поставки
НАЗНАЧЕНИЕ ЕДИНИЦЫ ИЗМЕНЕНИЯ (R_MEASURE_DS_MEDS)	НАЗНАЧЕНИЕ ЕДИНИЦЫ ИЗМЕНЕНИЯ	Standard	Справочник для примера	Пример данных назначения единиц измерения (3 строки)
ЕДИНИЦА ИЗМЕНЕНИЯ (S_EI_EIEI)	ЕДИНИЦА ИЗМЕРЕНИЯ	Standard	Справочник для примера	Пример данных единиц измерения (3 строки)
ЗАПРОС К БД (RQ_REQUEST_DB_RQDB)	Имя и заголовок запроса	Standard	Оперативная таблица инструментов	Нет
ЗАПРОС К БД ТЕКСТ (RQ_REQUEST_TXT_RQTX)	Текст запроса	Standard	Оперативная таблица инструментов	Нет
ВРЕМЕННАЯ ТАБЛИЦА (ACTIVE_TEMPORARY_TB_ATTB)	Управление временными таблицами	Standard	Оперативная таблица инструментов	Нет
ОБМЕН ДАННЫХ РАСПОЛОЖЕНИЕ (TUNING_DATASP_TEDP)	Расположение источника данных в обмене	Extended	Системный справочник разработчика инструментов	Возможные значения расположения данных для импорта (2 строки)
ОБМЕН ДАННЫХ ФОРМАТ (TUNING_DATASF_TEDF)	Формат источника данных в обмене	Extended	Системный справочник разработчика инструментов	Возможные форматы данных для импорта (7 строк)
С_ТИП ВВОДА (TUNING_EXCH_TYPE_TETP)	Тип ввода (операции) данных в обмене (экспорт/импорт)	Extended	Системный справочник разработчика инструментов	Возможные варианты обмена (3 строки)
НАСТРОЙКА ОБМЕНА (TUNING_EXCH_TEAA)	Имя настройки обмена	Extended	Оперативная таблица инструментов. Частично заполнена	Примеры настроек обмена. Таблица содержит заголовки настроек (3 строки)
НАСТРОЙКА ОБМЕНА ОБЪЕКТ (TUNING_EXCHNOV_TEOV)	Объекты настройки обмена	Extended	Оперативная таблица инструментов. Частично заполнена	Примеры настроек обмена. Таблица содержит описание объектов настроек (11 строк).
НАСТРОЙКА ОБМЕНА СВОЙСТВА ОБЪЕКТА (TUNING_EXCHPR_TEPR)	Свойства (параметры) настройки обмена	Extended	Оперативная таблица инструментов. Частично заполнена	Примеры настроек обмена. Таблица содержит описание

				свойств объектов настроек (6 строк)
НАСТРОЙКА БД ИСТОЧНИКА (TUNING_TB_SOURCE_TTSR)	Имя описания структуры БД источника обмена	Extended	Системный справочник разработчика инструментов	Нет
НАСТРОЙКА БД ИСТОЧНИКА ПОЛЕ (TUNING_SOURCE_REL_TSRS)	Поля описания структуры БД источника обмена	Extended	Системный справочник разработчика инструментов	Нет
ИМПОРТ ВНЕШНЯЯ ТАБЛИЦА (IMPORT_EXERTIONS_TEMP_IEXT)	Рабочая для импорта	Extended	Оперативная таблица инструментов	Нет
ИМПОРТ ВНЕШНЯЯ ТАБЛИЦА С BLOB ДАННЫМИ (TEMP_EXT_TB_BLOB_TMPX)	Рабочая для импорта	Extended	Оперативная таблица инструментов	Нет
ТИП СИСТЕМЫ КООРДИНАТ (COORDINATE_TYPE_CRTP)	Тип системы координат	Full	Системные справочники разработчика инструментов	Содержит варианты систем координат – прямоугольная, географическая (2 строки)
ГЕОДЕЗИЧЕСКИЙ ДАТУМ (COORDINATE_DT_STDT)	Геодезический датум	Full	Прикладной справочник инструментов	Содержит примеры геодезических датумов (2 строки)
ГРУППА СИСТЕМЫ КООРДИНАТ (COORDINATE_GR_STGR)	Группа системы координат	Full	Прикладной справочник инструментов	Содержит примеры групп систем координат (5 строк)
МЕТОД ВЫСОТНОЙ ПРИВЯЗКИ (METHOD_TIE_ALTIT_MTTA)	Метод высотной привязки	Full	Прикладной справочник инструментов	Содержит примеры методов высотной привязки (23 строки)
МЕТОД ПЛАНОВОЙ ПРИВЯЗКИ (METHOD_TIE_PLANE_MTPP)	Метод плановой привязки	Full	Прикладной справочник инструментов	Содержит примеры методов плановой привязки (23 строки)
СИСТЕМА ВЫСОТ (SYSTEM_HEIGHT_STMH)	Система высот	Full	Прикладной справочник инструментов	Содержит пример системы высот (1 строка)
С_ЗОНА КООРДИНАТ (COORDINATE_ZN_ORZN)	Зона координат	Full	Прикладной справочник инструментов	Содержит примеры зон координат (4 строки)
ФОРМУЛА ПЕРЕСЧЕТА (FORMULA_CONVERT_FRMA)	Формула пересчета координат	Full	Прикладной справочник инструментов	Нет
С_МАСШТАБ ОТОБРАЖЕНИЯ (R_SCALE_RSCL)	Масштаб отображения	Full	Прикладной справочник инструментов	Содержит примеры масштабов отображения (8 строк)
СИСТЕМА КООРДИНАТ (COORDINATE_STMC)	Система координат	Full	Прикладной справочник инструментов	Содержит примеры систем координат (6 строк)
КООРДИНАТА ОБЪЕКТА (UN_ORDINATEXUN_UOXY)	Координаты объекта	Full	Оперативная таблица инструментов	Нет
С_МК ВИД СЛОЯ (MM_LAYER_MMVR)	Вид слоя	Full	Системный справочник пользователя инструментов	Содержит примеры вида слоя на картах (2 строки)
С_МК ТОПОЛОГИЯ В СЛОЕ (S_TPL_LAY_MMTL)	Топология в слое	Full	Прикладной справочник инструментов	Нет
С_МК ПРАВИЛО ТОПОЛОГИИ СЛОЯ (S_RULE_TPL_LAY_MMLV)	Правило топологии ArcMap	Full	Системный справочник разработчика инструментов	Содержит правила топологии ArcMap (32 строки)
МК СЛОЙ (MM_LAYER_MMLR)	Имя (заголовок) слоя карты	Full	Прикладной справочник инструментов	Нет
МК ПОЛЕ СЛОЯ (MM_FL_LAYER_MMLF)	Свойства (поля) слоя карты	Full	Прикладной справочник инструментов	Нет
МК ПРОЕКТ (MM_PRJMAP_MMPM)	Проект карты	Full	Оперативная таблица инструментов	Нет
МК ПРОЕКТ ARCREADER (MM_PRJMAP_GISR_MMGR)	Проект карты для ARCREADER	Full	Оперативная таблица инструментов	Нет
МК ФРЕЙМ ПРОЕКТА (MM_PRJMAP_PF_MMPF)	Фрейм проекта карты	Full	Оперативная таблица инструментов	Нет
МК КОНТУР ПРОЕКТА (MM_PRJMAP_CN_MMCM)	Контур проекта карты	Full	Оперативная таблица инструментов	Нет
МК СЛОЙ ПРОЕКТА (MM_PRJMAP_LR_MMLP)	Слой проекта карты	Full	Оперативная таблица инструментов	Нет
МК СВОЙСТВО КЛАССА В СЛОЕ (MM_PRJMAP_LOB_MMLB)	Свойство класса в слое проекта карты	Full	Оперативная таблица инструментов	Нет
МК ОБЪЕКТ В СЛОЕ ПРОЕКТА (MM_PRJMAP_LRO_MMLO)	Объект в слое проекта карты	Full	Оперативная таблица инструментов	Нет

МК ОБЪЕКТ В СЛОЕ КООРДИНАТА (MM_PRJMAP_LRD_MMLD)	Координата объекта в слое проекта карты	Full	Оперативная таблица инструментов	Нет
PM_MENU	Меню проекта	Standard		Меню главной формы

Приложение 3. Состав и назначение привилегий на операции

В процессе создания нового проекта конфигуратор проекта записывает в базу данных привилегии на операции, которые необходимы для работы инструментальных средств. Ниже приведено назначение и состав этих привилегий. В зависимости от состава операций в проекте, некоторые из этих привилегий должны быть переданы конечному пользователю разработанной системы.

Привилегия	Назначение	Использование	Входит в комплект поставки
s_Login	Запуск задачи	Общесистемная	Base, Standard, Extended, Full
Call_Menu_Log	Пункт меню Журнал системы	Общесистемная	Base, Standard, Extended, Full
VIEW_DDOB	Просмотр дополнительных данных	Общесистемная	Base, Standard, Extended, Full
BUILD_DDOB	Добавление дополнительных данных	Общесистемная	Base, Standard, Extended, Full
CHAN_DDOB	Редактирование дополнительных данных	Общесистемная	Base, Standard, Extended, Full
COPY_DDOB	Копирование дополнительных данных	Общесистемная	Base, Standard, Extended, Full
DEL_DDOB	Удаление дополнительных данных	Общесистемная	Base, Standard, Extended, Full
R_R_MEASURE_DS_MEDS	Просмотр системного справочника "С НАЗНАЧЕНИЕ ЕИ"	Общесистемная	Base, Standard, Extended, Full
R_S_EI_EIEI	Просмотр справочника "С ЕДИНИЦА ИЗМЕРЕНИЯ"	Общесистемная	Base, Standard, Extended, Full
W_S_EI_EIEI	Редактирование справочника "С ЕДИНИЦА ИЗМЕРЕНИЯ"	Общесистемная	Base, Standard, Extended, Full
CALL_MENU_CHAN_PASS	Пункт меню Изменение и шифрация пароля общего пользователя	Общесистемная	Base, Standard, Extended, Full
R_OBJECTS_PROP_OBJP	Просмотр системного справочника СВОЙСТВО КЛАССА	Модули "Конструктор запросов", "Обмен данными", "Карты в среде ArcGis"	Standard, Extended, Full
R_CLASS_DESCRIPTION	Просмотр системного справочника ОПИСАНИЕ КЛАССА	Модули "Конструктор запросов", "Обмен данными", "Карты в среде ArcGis"	Standard, Extended, Full
Call_Graf_Constructor	Пункт меню "Конструктор схем"	Модуль "Генератор экранных форм"	Standard, Extended, Full
FORMS_VIEW_SCHEME	Просмотр схемы формы	Модуль "Генератор экранных форм"	Standard, Extended, Full
FORMS_VIEW_FILTERS	Смена ролей узлов графа	Модуль "Генератор экранных форм"	Standard, Extended, Full
FORMS_VIEW_CONSOLE	Просмотр консоли формы	Модуль "Генератор экранных форм"	Standard, Extended, Full
CALL_MENU_QUERYBUILDER	Пункт меню "Список запросов к БД"	Модуль "Конструктор запросов"	Standard, Extended, Full
VIEW_RQDB	Просмотр таблицы "ЗАПРОС К БД"	Модуль "Конструктор запросов"	Standard, Extended, Full

BUILD_RQDB	Добавление в таблицу "ЗАПРОС К БД"	Модуль "Конструктор запросов"	Standard, Extended, Full
CHAN_RQDB	Изменение таблицы "ЗАПРОС К БД"	Модуль "Конструктор запросов"	Standard, Extended, Full
COPY_RQDB	Копирование строки таблицы "ЗАПРОС К БД"	Модуль "Конструктор запросов"	Standard, Extended, Full
DEL_RQDB	Удаление из таблицы "ЗАПРОС К БД"	Модуль "Конструктор запросов"	Standard, Extended, Full
Call_Interpreter_Formula	Генератор отчетов	Модуль "Генератор отчетов"	Extended, Full
Call_Reports	Вызов отчетов	Модуль "Генератор отчетов"	Extended, Full
R_TUNING_EXCH_TYPE_ETP	Просмотр системного справочника "Тип обмена"	Модуль "Обмен данными"	Extended, Full
R_TUNING_DATASP_TEDP	Просмотр системного справочника "ОБМЕН ДАННЫХ РАСПОЛОЖЕНИЕ"	Модуль "Обмен данными"	Extended, Full
R_TUNING_DATASF_TEDF	Просмотр системного справочника "Формат данных обмена"	Модуль "Обмен данными"	Extended, Full
CALL_MENU_TUNING_EXCH_TEAA	Пункт меню "Обмен данными"	Модуль "Обмен данными"	Extended, Full
VIEW_TEAA	Просмотр таблицы "Настройка обмена"	Модуль "Обмен данными"	Extended, Full
BUILD_TEAA	Добавление в таблицу "Настройка обмена"	Модуль "Обмен данными"	Extended, Full
CHAN_TEAA	Редактирование таблицы "Настройка обмена"	Модуль "Обмен данными"	Extended, Full
COPY_TEAA	Копирование строк таблицы "Настройка обмена"	Модуль "Обмен данными"	Extended, Full
DEL_TEAA	Удаление из таблицы "Настройка обмена"	Модуль "Обмен данными"	Extended, Full
Call_Data_Exchange	Обмен данными. Запуск обмена для текущей настройки	Модуль "Обмен данными"	Extended, Full
R_R_MEASURE_DS_MEDS	Просмотр системного справочника "С_НАЗНАЧЕНИЕ ЕИ"	Модуль "Карты в среде ArcGis"	Full
R_S_EI_EIEI	Просмотр справочника "С_ЕДИНИЦА ИЗМЕРЕНИЯ"	Модуль "Карты в среде ArcGis"	Full
W_S_EI_EIEI	Редактирование справочника "С_ЕДИНИЦА ИЗМЕРЕНИЯ"	Модуль "Карты в среде ArcGis"	Full
R_COORDINATE_TYPE_C RTP	Просмотр системного справочника "ТИП СИСТЕМЫ КООРДИНАТ"	Модуль "Карты в среде ArcGis"	Full
R_SYSTEM_HEIGHT_STM H	Просмотр справочника "Система высот"	Модуль "Карты в среде ArcGis"	Full
R_R_SCALE_RSCL	Просмотр справочника "Масштаб отображения"	Модуль "Карты в среде ArcGis"	Full
R_METHOD_TIE_PLANE_M TTP	Просмотр справочника "МЕТОД ПЛАНОВОЙ ПРИВЯЗКИ"	Модуль "Карты в среде ArcGis"	Full
R_METHOD_TIE_ALTIT_M TTA	Просмотр справочника "МЕТОД ВЫСОТНОЙ ПРИВЯЗКИ"	Модуль "Карты в среде ArcGis"	Full
R_FORMULA_CONVERT_F RMA	Просмотр справочника "ФОРМУЛА ПЕРЕСЧЕТА"	Модуль "Карты в среде ArcGis"	Full
R_COORDINATE_ZN_ORZ N	Просмотр справочника "С_ЗОНА КООРДИНАТ"	Модуль "Карты в среде ArcGis"	Full
R_COORDINATE_STMC	Просмотр справочника "СИСТЕМА КООРДИНАТ"	Модуль "Карты в среде ArcGis"	Full
R_COORDINATE_GR_STG R	Просмотр справочника "ГРУППА СИСТЕМЫ КООРДИНАТ"	Модуль "Карты в среде ArcGis"	Full
R_COORDINATE_DT_STD T	Просмотр справочника "ГЕОДЕЗИЧЕСКИЙ ДАТУМ"	Модуль "Карты в среде ArcGis"	Full
W_SYSTEM_HEIGHT_STM H	Редактирование справочника "Система высот"	Модуль "Карты в среде ArcGis"	Full
W_R_SCALE_RSCL	Редактирование справочника "Масштаб отображения"	Модуль "Карты в среде ArcGis"	Full

W_METHOD_TIE_PLANE_MTPP	Редактирование справочника "МЕТОД ПЛАНОВОЙ ПРИВЯЗКИ"	Модуль "Карты в среде ArcGis"	Full
W_METHOD_TIE_ALTIT_MTTA	Редактирование справочника "МЕТОД ВЫСОТНОЙ ПРИВЯЗКИ"	Модуль "Карты в среде ArcGis"	Full
W_FORMULA_CONVERT_FRMA	Редактирование справочника "ФОРМУЛА ПЕРЕСЧЕТА"	Модуль "Карты в среде ArcGis"	Full
W_COORDINATE_ZN_ORZN	Редактирование справочника "С ЗОНА КООРДИНАТ"	Модуль "Карты в среде ArcGis"	Full
W_COORDINATE_STMC	Редактирование справочника "СИСТЕМА КООРДИНАТ"	Модуль "Карты в среде ArcGis"	Full
W_COORDINATE_GR_STGR	Редактирование справочника "ГРУППА СИСТЕМЫ КООРДИНАТ"	Модуль "Карты в среде ArcGis"	Full
W_COORDINATE_DT_STD T	Редактирование справочника "ГЕОДЕЗИЧЕСКИЙ ДАТУМ"	Модуль "Карты в среде ArcGis"	Full
R_MM_LAYER_MMVR	Просмотр системного справочника "С_МК ВИД СЛОЯ"	Модуль "Карты в среде ArcGis"	Full
R_S_RULE_TPL_LAY_MMVL	Просмотр системного справочника С_МК ПРАВИЛО ТОПОЛОГИИ СЛОЯ	Модуль "Карты в среде ArcGis"	Full
R_S_TPL_LAY_MMTL	Просмотр справочника "С_МК Топология в слое"	Модуль "Карты в среде ArcGis"	Full
W_S_TPL_LAY_MMTL	Редактирование справочника "С_МК Топология в слое"	Модуль "Карты в среде ArcGis"	Full
R_MM_LAYER_MMLR	Просмотр справочника "МК СЛОЙ"	Модуль "Карты в среде ArcGis"	Full
W_MM_LAYER_MMLR	Редактирование справочника "МК СЛОЙ"	Модуль "Карты в среде ArcGis"	Full
R_MM_FL_LAYER_MMLF	Просмотр справочника "МК ПОЛЕ СЛОЯ"	Модуль "Карты в среде ArcGis"	Full
W_MM_FL_LAYER_MMLF	Редактирование справочника "МК ПОЛЕ СЛОЯ"	Модуль "Карты в среде ArcGis"	Full
Call_Map_Project_List	Пункт меню "Список проектов карт"	Модуль "Карты в среде ArcGis"	Full
VIEW_MMPM	Просмотр таблицы "МК ПРОЕКТ"	Модуль "Карты в среде ArcGis"	Full
VIEW_MMPF	Просмотр таблицы МК ФРЕЙМ ПРОЕКТА	Модуль "Карты в среде ArcGis"	Full
VIEW_MMPL	Просмотр таблицы "МК СЛОЙ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
VIEW_MMLO	Просмотр таблицы "МК ОБЪЕКТ В СЛОЕ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
BUILD_MMPM	Добавление в таблицу "МК ПРОЕКТ"	Модуль "Карты в среде ArcGis"	Full
BUILD_MMPF	Добавление новой строки в таблицу МК ФРЕЙМ ПРОЕКТА	Модуль "Карты в среде ArcGis"	Full
BUILD_MMPL	Добавление в таблицу "МК СЛОЙ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
BUILD_MMLO	Добавление в таблицу "МК ОБЪЕКТ В СЛОЕ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
CHAN_MMPM	Изменение таблицы "МК ПРОЕКТ"	Модуль "Карты в среде ArcGis"	Full
CHAN_MMPF	Редактирование строки таблицы МК ФРЕЙМ ПРОЕКТА	Модуль "Карты в среде ArcGis"	Full
CHAN_MMPL	Изменение таблицы "МК СЛОЙ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
CHAN_MMLO	Изменение таблицы "МК ОБЪЕКТ В СЛОЕ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
COPY_MMPM	Копирование строки таблицы "МК ПРОЕКТ"	Модуль "Карты в среде ArcGis"	Full
COPY_MMPF	Копирование строки таблицы МК ФРЕЙМ ПРОЕКТА	Модуль "Карты в среде ArcGis"	Full
COPY_MMPL	Копирование строки таблицы "МК СЛОЙ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full

COPY_MMLO	Копирование строки таблицы "МК ОБЪЕКТ В СЛОЕ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
DEL_MMPM	Удаление из таблицы "МК ПРОЕКТ"	Модуль "Карты в среде ArcGis"	Full
DEL_MMPF	Удаление строки из таблицы МК ФРЕЙМ ПРОЕКТА	Модуль "Карты в среде ArcGis"	Full
DEL_MMPL	Удаление из таблицы "МК СЛОЙ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
DEL_MMLO	Удаление из таблицы "МК ОБЪЕКТ В СЛОЕ ПРОЕКТА"	Модуль "Карты в среде ArcGis"	Full
Temp_PRJ_MMPM	Действия с временными проектами карт	Модуль "Карты в среде ArcGis"	Full
Use_ArcReader	Передача объектов на карту с использованием ArcReader	Модуль "Карты в среде ArcGis"	Full
Use_ArcMap	Передача объектов на карту с использованием ArcMap	Модуль "Карты в среде ArcGis"	Full
ARCMAP_VIEW_CARD_PRJ	ArcMap. Просмотреть карточку проекта	Модуль "Карты в среде ArcGis"	Full
ARCMAP_VIEW_ALL_OBJ	ArcMap. Отобразить все объекты	Модуль "Карты в среде ArcGis"	Full
ARCMAP_VIEW	ArcMap. Просмотр информации об объектах	Модуль "Карты в среде ArcGis"	Full
ARCMAP_SAVE_SHP_FILES	ArcMap. Сохранить шейп-файлы	Модуль "Карты в среде ArcGis"	Full
ARCMAP_SAVE_PRJ_DESIGN	ArcMap. Сохранить оформление проекта	Модуль "Карты в среде ArcGis"	Full
ARCMAP_SAVE_PRJ_AS	ArcMap. Сохранить проект карты с новым именем	Модуль "Карты в среде ArcGis"	Full
ARCMAP_SAVE_PRJ	ArcMap. Сохранить проект	Модуль "Карты в среде ArcGis"	Full
ARCMAP_EDIT	ArcMap. Редактирование	Модуль "Карты в среде ArcGis"	Full
ARCMAP_DEL_LAYER_OBJ	ArcMap. Удалить объекты слоя из проекта	Модуль "Карты в среде ArcGis"	Full
ARCMAP_DEL_LAYER	ArcMap. Удалить слой из проекта	Модуль "Карты в среде ArcGis"	Full
ARCMAP_CENTROID_COORD	ArcMap. Записать координаты центроидов	Модуль "Карты в среде ArcGis"	Full
ARCMAP_ADD_OBJ_TO_LAYER	ArcMap. Добавить объекты в слой	Модуль "Карты в среде ArcGis"	Full
ARCMAP_ADD_LAYER	ArcMap. Добавить слой	Модуль "Карты в среде ArcGis"	Full

Приложение 4. Скрипты на создание демонстрационных таблиц для MS SQL Server и Oracle

Ниже приведены скрипты на создание демонстрационных таблиц для MS SQL Server и Oracle. В скриптах, в комментариях к таблицам и полям соответственно, присутствуют русские имена, поэтому сущности будут сгенерированы сразу с русскими именами таблиц и полей.

Скрипты для создание демонстрационных таблиц для MS SQL Server

```
USE [PM_INSTALL_DATA]
GO
```

```
/****** Object: Table [dbo].[SR_DETAIL_REMARK_SRDR]   Script Date: 07.06.2018 14:15:41 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[SR_DETAIL_REMARK_SRDR](
    [ID_SR_DETAIL_REMARK_SRDR] [int] NOT NULL,
    [ID_SR_REMARK_SRDR] [int] NOT NULL,
    [NUMBER_PP_SRDR] [int] NOT NULL,
    [DESCRIPTION_ACTION_SRDR] [varchar](2000) NOT NULL,
    [LINK_SRDR] [varchar](256) NULL,
    [VVOID_ID_CONTRACTOR_SRDR] [varchar](30) NOT NULL,
    [CHANGE_ID_CONTRACTOR_SRDR] [varchar](30) NULL,
    [DATE_INPUT_SRDR] [datetime] NOT NULL DEFAULT (getdate()),
    [DATE_CHANGE_SRDR] [datetime] NULL,
    CONSTRAINT [XPKSR_DETAIL_REMARK_SRDR] PRIMARY KEY NONCLUSTERED
(
    [ID_SR_DETAIL_REMARK_SRDR] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK1SR_DETAIL_REMARK_SRDR] UNIQUE NONCLUSTERED
(
    [ID_SR_REMARK_SRDR] ASC,
    [NUMBER_PP_SRDR] 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
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID_Детализация',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',
@level2type=N'COLUMN',@level2name=N'ID_SR_DETAIL_REMARK_SRDR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID_Замечание',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',
@level2type=N'COLUMN',@level2name=N'ID_SR_REMARK_SRDR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'№ п/п',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',
@level2type=N'COLUMN',@level2name=N'NUMBER_PP_SRDR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Описание последовательности действий',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',
@level2type=N'COLUMN',@level2name=N'DESCRIPTION_ACTION_SRDR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Ссылка на документ или скриншот',  
@level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',  
@level2type=N'COLUMN',@level2name=N'LINK_SRDR'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор ввода',  
@level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',  
@level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SRDR'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор изменения',  
@level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',  
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SRDR'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата ввода',  
@level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',  
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SRDR'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата изменения',  
@level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR',  
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SRDR'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=СЗ_ДЕТАЛИЗАЦИЯ  
ЗАМЕЧАНИЯ;Sequence=SEQ_SR_DETAIL_REMARK_SRDR', @level0type=N'SCHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_DETAIL_REMARK_SRDR'  
GO
```

```
USE [PM_INSTALL_DATA]  
GO
```

```
/***** Object: Table [dbo].[SR_PHYS_PERSON_SRPP] Script Date: 07.06.2018 14:16:35 *****/  
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON  
GO
```

```
SET ANSI_PADDING ON  
GO
```

```
CREATE TABLE [dbo].[SR_PHYS_PERSON_SRPP](  
    [ID_SR_PHYS_PERSON_SRPP] [int] NOT NULL,  
    [ID_SR_R_EXECUTIVE_SRPP] [int] NULL,  
    [DATE_BIRTH_SRPP] [datetime] NULL,  
    [EDUCATION_SRPP] [varchar](500) NULL,  
    [POST_SRPP] [varchar](100) NULL,  
    [PHONE_SRPP] [varchar](100) NULL,  
    [EMAIL_SRPP] [varchar](50) NULL,  
    [SKYPE_SRPP] [varchar](20) NULL,  
    [ICQ_SRPP] [varchar](20) NULL,  
    [OTHER_CONTACT_SRPP] [varchar](200) NULL,  
    [COMMENT_SRPP] [varchar](200) NULL,
```

```

[VVOD_ID_CONTRACTOR_SRPP] [varchar](30) NOT NULL,
[CHANGE_ID_CONTRACTOR_SRPP] [varchar](30) NULL,
[DATE_INPUT_SRPP] [datetime] NOT NULL DEFAULT (getdate()),
[DATE_CHANGE_SRPP] [datetime] NULL,
CONSTRAINT [XPKSR_PHYS_PERSON_SRPP] PRIMARY KEY NONCLUSTERED
(
    [ID_SR_PHYS_PERSON_SRPP] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [XAK1SR_PHYS_PERSON_SRPP] UNIQUE NONCLUSTERED
(
    [ID_SR_R_EXECUTIVE_SRPP] 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

```

```

EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'ID_Физическое лицо' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',
@level2type=N'COLUMN',@level2name=N'ID_SR_PHYS_PERSON_SRPP'
GO

```

```

EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'ID_Исполнитель' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',
@level2type=N'COLUMN',@level2name=N'ID_SR_R_EXECUTIVE_SRPP'
GO

```

```

EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата рождения' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',
@level2type=N'COLUMN',@level2name=N'DATE_BIRTH_SRPP'
GO

```

```

EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Образование' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',
@level2type=N'COLUMN',@level2name=N'EDUCATION_SRPP'
GO

```

```

EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Должность' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',
@level2type=N'COLUMN',@level2name=N'POST_SRPP'
GO

```

```

EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Контактный телефон' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',
@level2type=N'COLUMN',@level2name=N'PHONE_SRPP'
GO

```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Адрес электронной почты',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'EMAIL_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Skype',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'SKYPE_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'ICQ',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'ICQ_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Прочие контакты',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'OTHER_CONTACT_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Примечание',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'COMMENT_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор ввода',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор изменения',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата ввода',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата изменения',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP',  
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SRPP'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Физическое  
лицо;Sequence=SEQ_SR_PHYS_PERSON_SRPP', @level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_PHYS_PERSON_SRPP'  
GO
```

```
USE [PM_INSTALL_DATA]
```

GO

/***** Object: Table [dbo].[SR_R_CLIENT_SRCL] Script Date: 07.06.2018 14:16:45 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

```
CREATE TABLE [dbo].[SR_R_CLIENT_SRCL](
    [ID_SR_R_CLIENT_SRCL] [int] NOT NULL,
    [NAME_CLIENT_SRCL] [varchar](200) NOT NULL,
    [NAME_SHORT_SRCL] [varchar](50) NULL,
    [ADDRESS_SRCL] [varchar](500) NULL,
    [ADDRESS_FACT_SRCL] [varchar](500) NULL,
    [PHONE_SRCL] [varchar](100) NULL,
    [FAX_SRCL] [varchar](100) NULL,
    [EMAIL_SRCL] [varchar](50) NULL,
    [COMMENT_SRCL] [varchar](200) NULL,
    [VVOID_ID_CONTRACTOR_SRCL] [varchar](30) NOT NULL,
    [CHANGE_ID_CONTRACTOR_SRCL] [varchar](30) NULL,
    [DATE_INPUT_SRCL] [datetime] NOT NULL DEFAULT (getdate()),
    [DATE_CHANGE_SRCL] [datetime] NULL,
    CONSTRAINT [XPKSR_R_CLIENT_SRCL] PRIMARY KEY NONCLUSTERED
(
    [ID_SR_R_CLIENT_SRCL] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK1SR_R_CLIENT_SRCL] UNIQUE NONCLUSTERED
(
    [NAME_CLIENT_SRCL] 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

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'ID_Заказчик' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'ID_SR_R_CLIENT_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Наименование полное' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'NAME_CLIENT_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Наименование краткое' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'NAME_SHORT_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Юридический адрес' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'ADDRESS_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Фактический адрес' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'ADDRESS_FACT_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Телефон' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'PHONE_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Факс' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'FAX_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Адрес электронной почты' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'EMAIL_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Примечание' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'COMMENT_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор ввода' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор изменения' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата ввода' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата изменения' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL',
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SRCL'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@value=N'Caption=C3_C_3AKA3ЧИК;Sequence=SEQ_SR_R_CLIENT_SRCL' ,
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_CLIENT_SRCL'
GO
```

```
USE [PM_INSTALL_DATA]
GO
```

```
/***** Object: Table [dbo].[SR_R_EXECUTIVE_SREX] Script Date: 07.06.2018 14:16:54 *****/
SET ANSI_NULLS ON
```

GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

```
CREATE TABLE [dbo].[SR_R_EXECUTIVE_SREX](
    [ID_SR_R_EXECUTIVE_SREX] [int] NOT NULL,
    [NUMBER_PP_SREX] [int] NOT NULL,
    [FIO_SREX] [varchar](64) NOT NULL,
    [LOGIN_SREX] [varchar](30) NOT NULL,
    [LOGIN_IN_CHAT_SREX] [varchar](50) NULL,
    [VVOID_ID_CONTRACTOR_SREX] [varchar](30) NOT NULL,
    [CHANGE_ID_CONTRACTOR_SREX] [varchar](30) NULL,
    [DATE_INPUT_SREX] [datetime] NOT NULL DEFAULT (getdate()),
    [DATE_CHANGE_SREX] [datetime] NULL,
    CONSTRAINT [XPKSR_R_EXECUTIVE_SREX] PRIMARY KEY NONCLUSTERED
(
    [ID_SR_R_EXECUTIVE_SREX] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK1SR_R_EXECUTIVE_SREX] UNIQUE NONCLUSTERED
(
    [NUMBER_PP_SREX] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK2SR_R_EXECUTIVE_SREX] UNIQUE NONCLUSTERED
(
    [LOGIN_SREX] 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

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID_Исполнитель',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',
@level2type=N'COLUMN',@level2name=N'ID_SR_R_EXECUTIVE_SREX'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'№ п/п',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',
@level2type=N'COLUMN',@level2name=N'NUMBER_PP_SREX'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'ФИО',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',
@level2type=N'COLUMN',@level2name=N'FIO_SREX'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Логин',
@level0type=N'SHEMA',@level0name=N'dbo',
```



```
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',  
@level2type=N'COLUMN',@level2name=N'LOGIN_SREX'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Логин в рабочем чате',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',  
@level2type=N'COLUMN',@level2name=N'LOGIN_IN_CHAT_SREX'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор ввода',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',  
@level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SREX'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор изменения',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',  
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SREX'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата ввода',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',  
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SREX'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата изменения',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX',  
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SREX'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',  
@value=N'Caption=СЗ_С_ИСПОЛНИТЕЛЬ;Sequence=SEQ_SR_R_EXECUTIVE_SREX',  
@level0type=N'SHEMA',@level0name=N'dbo',  
@level1type=N'TABLE',@level1name=N'SR_R_EXECUTIVE_SREX'  
GO
```

```
USE [PM_INSTALL_DATA]  
GO
```

```
/***** Object: Table [dbo].[SR_R_PROJECT_SRPJ] Script Date: 07.06.2018 14:17:07 *****/  
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON  
GO
```

```
SET ANSI_PADDING ON  
GO
```

```
CREATE TABLE [dbo].[SR_R_PROJECT_SRPJ](  
    [ID_SR_R_PROJECT_SRPJ] [int] NOT NULL,  
    [NAME_PROJ_SRPJ] [varchar](20) NOT NULL,  
    [DESCRIPTION_SRPJ] [varchar](100) NULL,  
    [ID_SR_R_CLIENT_SRPJ] [int] NULL,  
    [ID_SR_PHYS_PERSON_SRPJ] [int] NULL,  
    [VVOID_ID_CONTRACTOR_SRPJ] [varchar](30) NOT NULL,
```

```
[CHANGE_ID_CONTRACTOR_SRPJ] [varchar](30) NULL,  
[DATE_INPUT_SRPJ] [datetime] NOT NULL DEFAULT (getdate()),  
[DATE_CHANGE_SRPJ] [datetime] NULL,  
CONSTRAINT [XPKSR_R_PROJECT_SRPJ] PRIMARY KEY NONCLUSTERED  
(  
    [ID_SR_R_PROJECT_SRPJ] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],  
CONSTRAINT [XAK1SR_R_PROJECT_SRPJ] UNIQUE NONCLUSTERED  
(  
    [NAME_PROJ_SRPJ] 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
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID_Проект',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'ID_SR_R_PROJECT_SRPJ'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Наименование проекта',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'NAME_PROJ_SRPJ'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Описание проекта',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'DESCRIPTION_SRPJ'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'ID Заказчик проекта',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'ID_SR_R_CLIENT_SRPJ'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'ID Контактное лицо',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'ID_SR_PHYS_PERSON_SRPJ'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор ввода',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SRPJ'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор изменения',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SRPJ'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата ввода',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',  
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SRPJ'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата изменения',
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ',
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SRPJ'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@value=N'Caption=C3_C_ППОЕКТ;Sequence=SEQ_SR_R_PROJECT_SRPJ',
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_PROJECT_SRPJ'
GO
```

```
USE [PM_INSTALL_DATA]
GO
```

```
/***** Object: Table [dbo].[SR_R_PROMPTNESS_SRPR] Script Date: 07.06.2018 14:17:19 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[SR_R_PROMPTNESS_SRPR](
    [ID_SR_R_PROMPTNESS_SRPR] [int] NOT NULL,
    [NUMBER_PP_SRPR] [int] NOT NULL,
    [NAME_SRPR] [varchar](6) NOT NULL,
    [DESCRIPTION_SRPR] [varchar](250) NULL,
    [VVOID_ID_CONTRACTOR_SRPR] [varchar](30) NOT NULL,
    [CHANGE_ID_CONTRACTOR_SRPR] [varchar](30) NULL,
    [DATE_INPUT_SRPR] [datetime] NOT NULL DEFAULT (getdate()),
    [DATE_CHANGE_SRPR] [datetime] NULL,
    CONSTRAINT [XPKSR_R_PROMPTNESS_SRPR] PRIMARY KEY NONCLUSTERED
(
    [ID_SR_R_PROMPTNESS_SRPR] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK1SR_R_PROMPTNESS_SRPR] UNIQUE NONCLUSTERED
(
    [NUMBER_PP_SRPR] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK2SR_R_PROMPTNESS_SRPR] UNIQUE NONCLUSTERED
(
    [NAME_SRPR] 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
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID_Срочность',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'ID_SR_R_PROMPTNESS_SRPR'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'№ п/п',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'NUMBER_PP_SRPR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Наименование',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'NAME_SRPR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Описание',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'DESCRIPTION_SRPR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор ввода',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SRPR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор изменения',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SRPR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата ввода',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SRPR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата изменения',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR',
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SRPR'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@value=N'Caption=C3_C_СПОЧНОСТЬ;Sequence=SEQ_SR_R_PROMPTNESS_SRPR',
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'SR_R_PROMPTNESS_SRPR'
GO
```

```
USE [PM_INSTALL_DATA]
GO
```

```
/****** Object: Table [dbo].[SR_R_STATUS_SRST] Script Date: 07.06.2018 14:17:31 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[SR_R_STATUS_SRST](
    [ID_SR_R_STATUS_SRST] [int] NOT NULL,
    [NUMBER_PP_SRST] [int] NOT NULL,
    [NAME_STATUS_SRST] [varchar](15) NOT NULL,
    [COMMENT_SRST] [varchar](250) NULL,
    [VVOID_ID_CONTRACTOR_SRST] [varchar](30) NOT NULL,
    [CHANGE_ID_CONTRACTOR_SRST] [varchar](30) NULL,
    [DATE_INPUT_SRST] [datetime] NOT NULL DEFAULT (getdate()),
    [DATE_CHANGE_SRST] [datetime] NULL,
    CONSTRAINT [XPKSR_R_STATUS_SRST] PRIMARY KEY NONCLUSTERED
(
    [ID_SR_R_STATUS_SRST] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK1SR_R_STATUS_SRST] UNIQUE NONCLUSTERED
(
    [NUMBER_PP_SRST] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [XAK2SR_R_STATUS_SRST] UNIQUE NONCLUSTERED
(
    [NAME_STATUS_SRST] 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
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID_Статус',
    @level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
    @level2type=N'COLUMN',@level2name=N'ID_SR_R_STATUS_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'№ п/п',
    @level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
    @level2type=N'COLUMN',@level2name=N'NUMBER_PP_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Наименование статуса',
    @level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
    @level2type=N'COLUMN',@level2name=N'NAME_STATUS_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Примечание',
    @level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
    @level2type=N'COLUMN',@level2name=N'COMMENT_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор ввода',
    @level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
    @level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Автор изменения' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата ввода' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Дата изменения' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST',
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SRST'
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',
@value=N'Caption=C3_C_CTATYC;Sequence=SEQ_SR_R_STATUS_SRST' ,
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_R_STATUS_SRST'
GO
```

```
USE [PM_INSTALL_DATA]
GO
```

```
/***** Object: Table [dbo].[SR_REMARK_SRRM] Script Date: 07.06.2018 14:17:44 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[SR_REMARK_SRRM](
    [ID_SR_REMARK_SRRM] [int] NOT NULL,
    [ID_SR_R_PROJECT_SRRM] [int] NOT NULL,
    [DATE_CREATE_SRRM] [datetime] NOT NULL DEFAULT (getdate()),
    [NUMBER_PP_SRRM] [int] NOT NULL,
    [ID_FROM_WHOM_SRRM] [int] NOT NULL,
    [ID_WHOM_SRRM] [int] NULL,
    [ID_SR_R_PROMPTNESS_SRRM] [int] NULL,
    [NAME_DB_SRRM] [varchar](20) NULL,
    [PROGRAM_SRRM] [varchar](50) NULL,
    [PLACE_ERROR_SRRM] [varchar](250) NULL,
    [REMARK_SRRM] [varchar](4000) NOT NULL,
    [LINK_SRRM] [varchar](256) NULL,
    [COMMENT_EXECUT_SRRM] [varchar](4000) NULL,
    [DATE_COMPLETE_SRRM] [datetime] NULL,
    [DATE_EXECUTE_SRRM] [datetime] NULL,
    [DATE_CHECKED_SRRM] [datetime] NULL,
    [ID_SR_R_STATUS_SRRM] [int] NOT NULL,
    [PARENT_REMARK_SRRM] [int] NULL,
    [VVOD_ID_CONTRACTOR_SRRM] [varchar](30) NOT NULL,
    [CHANGE_ID_CONTRACTOR_SRRM] [varchar](30) NULL,
    [DATE_INPUT_SRRM] [datetime] NOT NULL DEFAULT (getdate()),
    [DATE_CHANGE_SRRM] [datetime] NULL,
    CONSTRAINT [XPKS_SR_REMARK_SRRM] PRIMARY KEY NONCLUSTERED
(
    [ID_SR_REMARK_SRRM] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],  
CONSTRAINT [XAK1SR_REMARK_SRRM] UNIQUE NONCLUSTERED  
(  
    [ID_SR_R_PROJECT_SRRM] ASC,  
    [NUMBER_PP_SRRM] ASC,  
    [DATE_CREATE_SRRM] 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

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID задания',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'ID_SR_REMARK_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID проекта',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'ID_SR_R_PROJECT_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Дата создания',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'DATE_CREATE_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=№ п/п',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'NUMBER_PP_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=От кого',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'ID_FROM_WHOM_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Кому',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'ID_WHOM_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID срочности',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'ID_SR_R_PROMPTNESS_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=База данных',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'NAME_DB_SRRM'
```

GO

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Программа',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'PROGRAM_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Место возникновения ошибки',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'PLACE_ERROR_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Текст задания',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'REMARK_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Ссылка на документ или скриншот',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'LINK_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Комментарий исполнителя',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'COMMENT_EXECUT_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Плановый срок выполнения',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'DATE_COMPLETE_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Дата выполнения',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'DATE_EXECUTE_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Дата приемки',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'DATE_CHECKED_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=ID статус',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'ID_SR_R_STATUS_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Замечание-родитель',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'PARENT_REMARK_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Автор ввода',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'VVOID_ID_CONTRACTOR_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Автор изменения',  
@level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'CHANGE_ID_CONTRACTOR_SRRM'  
GO
```



```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Дата ввода',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'DATE_INPUT_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment', @value=N'Caption=Дата изменения',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM',  
@level2type=N'COLUMN',@level2name=N'DATE_CHANGE_SRRM'  
GO
```

```
EXEC sys.sp_addextendedproperty @name=N'Comment',  
@value=N'Caption=C3_3АМЕЧАНИЕ;Sequence=SEQ_SR_REMARK_SRRM',  
@level0type=N'SHEMA',@level0name=N'dbo', @level1type=N'TABLE',@level1name=N'SR_REMARK_SRRM'  
GO
```

```
ALTER TABLE [dbo].[SR_DETAIL_REMARK_SRDR] WITH CHECK ADD CONSTRAINT [R_2840] FOREIGN  
KEY([ID_SR_REMARK_SRDR])  
REFERENCES [dbo].[SR_REMARK_SRRM] ([ID_SR_REMARK_SRRM])  
GO
```

```
ALTER TABLE [dbo].[SR_DETAIL_REMARK_SRDR] CHECK CONSTRAINT [R_2840]  
GO
```

```
ALTER TABLE [dbo].[SR_PHYS_PERSON_SRPP] WITH CHECK ADD CONSTRAINT [R_2849] FOREIGN  
KEY([ID_SR_R_EXECUTIVE_SRPP])  
REFERENCES [dbo].[SR_R_EXECUTIVE_SREX] ([ID_SR_R_EXECUTIVE_SREX])  
GO
```

```
ALTER TABLE [dbo].[SR_PHYS_PERSON_SRPP] CHECK CONSTRAINT [R_2849]  
GO
```

```
ALTER TABLE [dbo].[SR_R_PROJECT_SRPJ] WITH CHECK ADD CONSTRAINT [R_2848] FOREIGN  
KEY([ID_SR_PHYS_PERSON_SRPJ])  
REFERENCES [dbo].[SR_PHYS_PERSON_SRPP] ([ID_SR_PHYS_PERSON_SRPP])  
GO
```

```
ALTER TABLE [dbo].[SR_R_PROJECT_SRPJ] CHECK CONSTRAINT [R_2848]  
GO
```

```
ALTER TABLE [dbo].[SR_R_PROJECT_SRPJ] WITH CHECK ADD CONSTRAINT [R_2850] FOREIGN  
KEY([ID_SR_R_CLIENT_SRPJ])  
REFERENCES [dbo].[SR_R_CLIENT_SRCL] ([ID_SR_R_CLIENT_SRCL])  
GO
```

```
ALTER TABLE [dbo].[SR_R_PROJECT_SRPJ] CHECK CONSTRAINT [R_2850]  
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] WITH CHECK ADD CONSTRAINT [R_2835] FOREIGN  
KEY([ID_SR_R_PROJECT_SRRM])  
REFERENCES [dbo].[SR_R_PROJECT_SRPJ] ([ID_SR_R_PROJECT_SRPJ])  
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] CHECK CONSTRAINT [R_2835]  
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] WITH CHECK ADD CONSTRAINT [R_2836] FOREIGN  
KEY([ID_FROM_WHOM_SRRM])  
REFERENCES [dbo].[SR_R_EXECUTIVE_SREX] ([ID_SR_R_EXECUTIVE_SREX])  
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] CHECK CONSTRAINT [R_2836]
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] WITH CHECK ADD CONSTRAINT [R_2837] FOREIGN
KEY([ID_WHOM_SRRM])
REFERENCES [dbo].[SR_R_EXECUTIVE_SREX] ([ID_SR_R_EXECUTIVE_SREX])
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] CHECK CONSTRAINT [R_2837]
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] WITH CHECK ADD CONSTRAINT [R_2838] FOREIGN
KEY([ID_SR_R_PROMPTNESS_SRRM])
REFERENCES [dbo].[SR_R_PROMPTNESS_SRPR] ([ID_SR_R_PROMPTNESS_SRPR])
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] CHECK CONSTRAINT [R_2838]
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] WITH CHECK ADD CONSTRAINT [R_2839] FOREIGN
KEY([ID_SR_R_STATUS_SRRM])
REFERENCES [dbo].[SR_R_STATUS_SRST] ([ID_SR_R_STATUS_SRST])
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] CHECK CONSTRAINT [R_2839]
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] WITH CHECK ADD CONSTRAINT [R_2841] FOREIGN
KEY([PARENT_REMARK_SRRM])
REFERENCES [dbo].[SR_REMARK_SRRM] ([ID_SR_REMARK_SRRM])
GO
```

```
ALTER TABLE [dbo].[SR_REMARK_SRRM] CHECK CONSTRAINT [R_2841]
GO
```

Скрипты для создание демонстрационных таблиц для Oracle

```
CREATE TABLE PM_DBA.SR_R_CLIENT_SRCL
(
  ID_SR_R_CLIENT_SRCL      NUMBER(8)      NOT NULL,
  NAME_CLIENT_SRCL        VARCHAR2(200 BYTE) NOT NULL,
  NAME_SHORT_SRCL         VARCHAR2(50 BYTE),
  ADDRESS_SRCL            VARCHAR2(500 BYTE),
  ADDRESS_FACT_SRCL       VARCHAR2(500 BYTE),
  PHONE_SRCL              VARCHAR2(100 BYTE),
  FAX_SRCL                VARCHAR2(100 BYTE),
  EMAIL_SRCL              VARCHAR2(50 BYTE),
  COMMENT_SRCL            VARCHAR2(200 BYTE),
  VVOD_ID_CONTRACTOR_SRCL VARCHAR2(30 BYTE) NOT NULL,
  CHANGE_ID_CONTRACTOR_SRCL VARCHAR2(30 BYTE),
  DATE_INPUT_SRCL         DATE            DEFAULT sysdate      NOT NULL,
  DATE_CHANGE_SRCL        DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;
```

```
COMMENT ON TABLE PM_DBA.SR_R_CLIENT_SRCL IS
'Caption=СЗ_С_ЗАКАЗЧИК;Sequence=SEQ_SR_R_CLIENT_SRCL';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.ID_SR_R_CLIENT_SRCL IS 'ID_Заказчик';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.NAME_CLIENT_SRCL IS 'Наименование полное';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.NAME_SHORT_SRCL IS 'Наименование краткое';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.ADDRESS_SRCL IS 'Юридический адрес';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.ADDRESS_FACT_SRCL IS 'Фактический адрес';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.PHONE_SRCL IS 'Телефон';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.FAX_SRCL IS 'Факс';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.EMAIL_SRCL IS 'Адрес электронной почты';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.COMMENT_SRCL IS 'Примечание';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.VVOD_ID_CONTRACTOR_SRCL IS 'Автор ввода';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.CHANGE_ID_CONTRACTOR_SRCL IS 'Автор
изменения';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.DATE_INPUT_SRCL IS 'Дата ввода';

COMMENT ON COLUMN PM_DBA.SR_R_CLIENT_SRCL.DATE_CHANGE_SRCL IS 'Дата изменения';


CREATE TABLE PM_DBA.SR_R_EXECUTIVE_SREX
(
  ID_SR_R_EXECUTIVE_SREX  NUMBER(8)      NOT NULL,
  NUMBER_PP_SREX          NUMBER(8)      NOT NULL,
  FIO_SREX                VARCHAR2(64 BYTE) NOT NULL,
  LOGIN_SREX              VARCHAR2(30 BYTE) NOT NULL,
  LOGIN_IN_CHAT_SREX      VARCHAR2(50 BYTE),
  VVOD_ID_CONTRACTOR_SREX VARCHAR2(30 BYTE) NOT NULL,
  CHANGE_ID_CONTRACTOR_SREX VARCHAR2(30 BYTE),
  DATE_INPUT_SREX         DATE           DEFAULT sysdate      NOT NULL,
  DATE_CHANGE_SREX        DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;

COMMENT ON TABLE PM_DBA.SR_R_EXECUTIVE_SREX IS
'Caption=СЗ_С_ИСПОЛНИТЕЛЬ;Sequence=SEQ_SR_R_EXECUTIVE_SREX';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.ID_SR_R_EXECUTIVE_SREX IS
'ID_Исполнитель';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.NUMBER_PP_SREX IS 'Caption=№ п/п';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.FIO_SREX IS 'ФИО';
```

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.LOGIN_SREX IS 'Логин';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.LOGIN_IN_CHAT_SREX IS 'Логин в рабочем чате';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.VVOD_ID_CONTRACTOR_SREX IS 'Автор ввода';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.CHANGE_ID_CONTRACTOR_SREX IS 'Автор изменения';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.DATE_INPUT_SREX IS 'Дата ввода';

COMMENT ON COLUMN PM_DBA.SR_R_EXECUTIVE_SREX.DATE_CHANGE_SREX IS 'Дата изменения';

```
CREATE TABLE PM_DBA.SR_R_PROMPTNESS_SRPR
(
  ID_SR_R_PROMPTNESS_SRPR  NUMBER(8)      NOT NULL,
  NUMBER_PP_SRPR           NUMBER(8)      NOT NULL,
  NAME_SRPR                VARCHAR2(6 BYTE) NOT NULL,
  DESCRIPTION_SRPR         VARCHAR2(250 BYTE),
  VVOD_ID_CONTRACTOR_SRPR  VARCHAR2(30 BYTE) NOT NULL,
  CHANGE_ID_CONTRACTOR_SRPR VARCHAR2(30 BYTE),
  DATE_INPUT_SRPR          DATE            DEFAULT sysdate      NOT NULL,
  DATE_CHANGE_SRPR         DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;
```

COMMENT ON TABLE PM_DBA.SR_R_PROMPTNESS_SRPR IS
'Caption=C3_C_СРОЧНОСТЬ;Sequence=SEQ_SR_R_PROMPTNESS_SRPR';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.ID_SR_R_PROMPTNESS_SRPR IS
'ID_Срочность';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.NUMBER_PP_SRPR IS 'Caption=№ п/п';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.NAME_SRPR IS 'Наименование';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.DESRIPTION_SRPR IS 'Описание';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.VVOD_ID_CONTRACTOR_SRPR IS 'Автор ввода';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.CHANGE_ID_CONTRACTOR_SRPR IS
'Автор изменения';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.DATE_INPUT_SRPR IS 'Дата ввода';

COMMENT ON COLUMN PM_DBA.SR_R_PROMPTNESS_SRPR.DATE_CHANGE_SRPR IS 'Дата изменения';

```
CREATE TABLE PM_DBA.SR_R_STATUS_SRST
```

```
(
  ID_SR_R_STATUS_SRST      NUMBER(8)      NOT NULL,
  NUMBER_PP_SRST           NUMBER(8)      NOT NULL,
  NAME_STATUS_SRST         VARCHAR2(15 BYTE) NOT NULL,
  COMMENT_SRST             VARCHAR2(250 BYTE),
  VVOD_ID_CONTRACTOR_SRST  VARCHAR2(30 BYTE) NOT NULL,
  CHANGE_ID_CONTRACTOR_SRST VARCHAR2(30 BYTE),
  DATE_INPUT_SRST          DATE            DEFAULT sysdate      NOT NULL,
  DATE_CHANGE_SRST         DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;

COMMENT ON TABLE PM_DBA.SR_R_STATUS_SRST IS
'Caption=C3_C_CTATYC;Sequence=SEQ_SR_R_STATUS_SRST';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.ID_SR_R_STATUS_SRST IS 'ID_Статус';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.NUMBER_PP_SRST IS 'Caption=№ п/п';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.NAME_STATUS_SRST IS 'Наименование статуса';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.COMMENT_SRST IS 'Примечание';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.VVOD_ID_CONTRACTOR_SRST IS 'Автор ввода';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.CHANGE_ID_CONTRACTOR_SRST IS 'Автор
изменения';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.DATE_INPUT_SRST IS 'Дата ввода';

COMMENT ON COLUMN PM_DBA.SR_R_STATUS_SRST.DATE_CHANGE_SRST IS 'Дата изменения';

CREATE INDEX PM_DBA.XAK1SR_R_CLIENT_SRCL ON PM_DBA.SR_R_CLIENT_SRCL
(NAME_CLIENT_SRCL)
LOGGING
NOPARALLEL;

CREATE INDEX PM_DBA.XAK1SR_R_EXECUTIVE_SREX ON PM_DBA.SR_R_EXECUTIVE_SREX
(NUMBER_PP_SREX)
LOGGING
NOPARALLEL;

CREATE INDEX PM_DBA.XAK1SR_R_PROMPTNESS_SRPR ON PM_DBA.SR_R_PROMPTNESS_SRPR
(NUMBER_PP_SRPR)
LOGGING
NOPARALLEL;

CREATE INDEX PM_DBA.XAK1SR_R_STATUS_SRST ON PM_DBA.SR_R_STATUS_SRST
(NUMBER_PP_SRST)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX PM_DBA.XAK2SR_R_EXECUTIVE_SREX ON PM_DBA.SR_R_EXECUTIVE_SREX  
(LOGIN_SREX)  
LOGGING  
NOPARALLEL;
```

```
CREATE INDEX PM_DBA.XAK2SR_R_PROMPTNESS_SRPR ON PM_DBA.SR_R_PROMPTNESS_SRPR  
(NAME_SRPR)  
LOGGING  
NOPARALLEL;
```

```
CREATE INDEX PM_DBA.XAK2SR_R_STATUS_SRST ON PM_DBA.SR_R_STATUS_SRST  
(NAME_STATUS_SRST)  
LOGGING  
NOPARALLEL;
```

```
CREATE UNIQUE INDEX PM_DBA.XPKSR_R_CLIENT_SRCL ON PM_DBA.SR_R_CLIENT_SRCL  
(ID_SR_R_CLIENT_SRCL)  
LOGGING  
NOPARALLEL;
```

```
CREATE UNIQUE INDEX PM_DBA.XPKSR_R_EXECUTIVE_SREX ON PM_DBA.SR_R_EXECUTIVE_SREX  
(ID_SR_R_EXECUTIVE_SREX)  
LOGGING  
NOPARALLEL;
```

```
CREATE UNIQUE INDEX PM_DBA.XPKSR_R_PROMPTNESS_SRPR ON  
PM_DBA.SR_R_PROMPTNESS_SRPR  
(ID_SR_R_PROMPTNESS_SRPR)  
LOGGING  
NOPARALLEL;
```

```
CREATE UNIQUE INDEX PM_DBA.XPKSR_R_STATUS_SRST ON PM_DBA.SR_R_STATUS_SRST  
(ID_SR_R_STATUS_SRST)  
LOGGING  
NOPARALLEL;
```

```
CREATE TABLE PM_DBA.SR_PHYS_PERSON_SRPP  
(  
  ID_SR_PHYS_PERSON_SRPP  NUMBER(8)      NOT NULL,  
  ID_SR_R_EXECUTIVE_SRPP  NUMBER(8),  
  DATE_BIRTH_SRPP         DATE,  
  EDUCATION_SRPP          VARCHAR2(500 BYTE),  
  POST_SRPP               VARCHAR2(100 BYTE),  
  PHONE_SRPP              VARCHAR2(100 BYTE),  
  EMAIL_SRPP              VARCHAR2(50 BYTE),  
  SKYPE_SRPP              VARCHAR2(20 BYTE),  
  ICQ_SRPP                VARCHAR2(20 BYTE),  
  OTHER_CONTACT_SRPP      VARCHAR2(200 BYTE),  
  COMMENT_SRPP            VARCHAR2(200 BYTE),  
  VVOD_ID_CONTRACTOR_SRPP VARCHAR2(30 BYTE) NOT NULL,  
  CHANGE_ID_CONTRACTOR_SRPP VARCHAR2(30 BYTE),
```

```

DATE_INPUT_SRPP      DATE      DEFAULT sysdate      NOT NULL,
DATE_CHANGE_SRPP     DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;

COMMENT ON TABLE PM_DBA.SR_PHYS_PERSON_SRPP IS 'Caption=Физическое
лицо;Sequence=SEQ_SR_PHYS_PERSON_SRPP';

COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.ID_SR_PHYS_PERSON_SRPP IS
'ID_Физическое лицо';

COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.ID_SR_R_EXECUTIVE_SRPP IS
'ID_Исполнитель';

COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.DATE_BIRTH_SRPP IS 'Дата рождения';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.EDUCATION_SRPP IS 'Образование';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.POST_SRPP IS 'Должность';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.PHONE_SRPP IS 'Контактный телефон';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.EMAIL_SRPP IS 'Адрес электронной почты';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.SKYPE_SRPP IS 'Skype';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.ICQ_SRPP IS 'ICQ';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.OTHER_CONTACT_SRPP IS 'Прочие
контакты';

COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.COMMENT_SRPP IS 'Примечание';

COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.VVOD_ID_CONTRACTOR_SRPP IS 'Автор
ввода';

COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.CHANGE_ID_CONTRACTOR_SRPP IS 'Автор
изменения';

COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.DATE_INPUT_SRPP IS 'Дата ввода';
COMMENT ON COLUMN PM_DBA.SR_PHYS_PERSON_SRPP.DATE_CHANGE_SRPP IS 'Дата изменения';

CREATE TABLE PM_DBA.SR_R_PROJECT_SRPJ
(
ID_SR_R_PROJECT_SRPJ      NUMBER(8)      NOT NULL,
NAME_PROJ_SRPJ            VARCHAR2(20 BYTE) NOT NULL,
DESCRIPTION_SRPJ          VARCHAR2(100 BYTE),
ID_SR_R_CLIENT_SRPJ       NUMBER(8),
ID_SR_PHYS_PERSON_SRPJ    NUMBER(8),
VVOD_ID_CONTRACTOR_SRPJ   VARCHAR2(30 BYTE) NOT NULL,
CHANGE_ID_CONTRACTOR_SRPJ VARCHAR2(30 BYTE),
DATE_INPUT_SRPJ           DATE            DEFAULT sysdate      NOT NULL,
DATE_CHANGE_SRPJ          DATE

```

```
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;

COMMENT ON TABLE PM_DBA.SR_R_PROJECT_SRPJ IS
'Caption=C3_C_ППОЕКТ;Sequence=SEQ_SR_R_PROJECT_SRPJ';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.ID_SR_R_PROJECT_SRPJ IS 'ID_Проект';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.NAME_PROJ_SRPJ IS 'Наименование проекта';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.DESCRPTION_SRPJ IS 'Описание проекта';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.ID_SR_R_CLIENT_SRPJ IS 'Caption=ID Заказчик
проекта';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.ID_SR_PHYS_PERSON_SRPJ IS 'Caption=ID
Контактное лицо';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.VVOD_ID_CONTRACTOR_SRPJ IS 'Автор ввода';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.CHANGE_ID_CONTRACTOR_SRPJ IS 'Автор
изменения';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.DATE_INPUT_SRPJ IS 'Дата ввода';

COMMENT ON COLUMN PM_DBA.SR_R_PROJECT_SRPJ.DATE_CHANGE_SRPJ IS 'Дата изменения';


CREATE INDEX PM_DBA.XAK1SR_R_PROJECT_SRPJ ON PM_DBA.SR_R_PROJECT_SRPJ
(NAME_PROJ_SRPJ)
LOGGING
NOPARALLEL;


CREATE INDEX PM_DBA.XIF1SR_PHYS_PERSON_SRPP ON PM_DBA.SR_PHYS_PERSON_SRPP
(ID_SR_R_EXECUTIVE_SRPP)
LOGGING
NOPARALLEL;


CREATE INDEX PM_DBA.XIF1SR_R_PROJECT_SRPJ ON PM_DBA.SR_R_PROJECT_SRPJ
(ID_SR_PHYS_PERSON_SRPJ)
LOGGING
NOPARALLEL;


CREATE INDEX PM_DBA.XIF2SR_R_PROJECT_SRPJ ON PM_DBA.SR_R_PROJECT_SRPJ
(ID_SR_R_CLIENT_SRPJ)
LOGGING
NOPARALLEL;


CREATE UNIQUE INDEX PM_DBA.XPKSR_PHYS_PERSON_SRPP ON PM_DBA.SR_PHYS_PERSON_SRPP
(ID_SR_PHYS_PERSON_SRPP)
LOGGING
```


NOPARALLEL;

```
CREATE UNIQUE INDEX PM_DBA.XPKSR_R_PROJECT_SRPJ ON PM_DBA.SR_R_PROJECT_SRPJ
(ID_SR_R_PROJECT_SRPJ)
LOGGING
NOPARALLEL;
```

```
CREATE TABLE PM_DBA.SR_REMARK_SRRM
(
  ID_SR_REMARK_SRRM      NUMBER(8)      NOT NULL,
  ID_SR_R_PROJECT_SRRM   NUMBER(8)      NOT NULL,
  DATE_CREATE_SRRM       DATE           DEFAULT sysdate      NOT NULL,
  NUMBER_PP_SRRM         NUMBER(8)      NOT NULL,
  ID_FROM_WHOM_SRRM      NUMBER(8)      NOT NULL,
  ID_WHOM_SRRM           NUMBER(8),
  ID_SR_R_PROMPTNESS_SRRM NUMBER(8),
  NAME_DB_SRRM           VARCHAR2(20 BYTE),
  PROGRAM_SRRM           VARCHAR2(50 BYTE),
  PLACE_ERROR_SRRM       VARCHAR2(250 BYTE),
  REMARK_SRRM            VARCHAR2(4000 BYTE) NOT NULL,
  LINK_SRRM              VARCHAR2(256 BYTE),
  COMMENT_EXECUT_SRRM    VARCHAR2(4000 BYTE),
  DATE_COMPLETE_SRRM     DATE,
  DATE_EXECUTE_SRRM      DATE,
  DATE_CHECKED_SRRM      DATE,
  ID_SR_R_STATUS_SRRM    NUMBER(8)      NOT NULL,
  PARENT_REMARK_SRRM     NUMBER(8),
  VVOD_ID_CONTRACTOR_SRRM VARCHAR2(30 BYTE) NOT NULL,
  CHANGE_ID_CONTRACTOR_SRRM VARCHAR2(30 BYTE),
  DATE_INPUT_SRRM        DATE           DEFAULT sysdate      NOT NULL,
  DATE_CHANGE_SRRM       DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;
```

```
COMMENT ON TABLE PM_DBA.SR_REMARK_SRRM IS
'Caption=СЗ_ЗАМЕЧАНИЕ;Sequence=SEQ_SR_REMARK_SRRM';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.ID_SR_REMARK_SRRM IS 'Caption=ID задания';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.ID_SR_R_PROJECT_SRRM IS 'Caption=ID проекта';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.DATE_CREATE_SRRM IS 'Дата создания';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.NUMBER_PP_SRRM IS 'Caption=№ п/п';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.ID_FROM_WHOM_SRRM IS 'От кого';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.ID_WHOM_SRRM IS 'Кому';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.ID_SR_R_PROMPTNESS_SRRM IS 'Caption=ID
срочности';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.NAME_DB_SRRM IS 'База данных';
```

```
COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.PROGRAM_SRRM IS 'Программа';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.PLACE_ERROR_SRRM IS 'Место возникновения
ошибки';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.REMARK_SRRM IS 'Caption=Текст задания';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.LINK_SRRM IS 'Ссылка на документ или скриншот';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.COMMENT_EXECUT_SRRM IS 'Комментарий
исполнителя';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.DATE_COMPLETE_SRRM IS 'Плановый срок
выполнения';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.DATE_EXECUTE_SRRM IS 'Дата выполнения';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.DATE_CHECKED_SRRM IS 'Дата приемки';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.ID_SR_R_STATUS_SRRM IS 'Caption=ID статус';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.PARENT_REMARK_SRRM IS 'Замечание-родитель';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.VVOD_ID_CONTRACTOR_SRRM IS 'Автор ввода';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.CHANGE_ID_CONTRACTOR_SRRM IS 'Автор
изменения';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.DATE_INPUT_SRRM IS 'Дата ввода';

COMMENT ON COLUMN PM_DBA.SR_REMARK_SRRM.DATE_CHANGE_SRRM IS 'Дата изменения';


CREATE INDEX PM_DBA.XAK1SR_REMARK_SRRM ON PM_DBA.SR_REMARK_SRRM
(ID_SR_R_PROJECT_SRRM, NUMBER_PP_SRRM, DATE_CREATE_SRRM)
LOGGING
NOPARALLEL;


CREATE INDEX PM_DBA.XIF1SR_REMARK_SRRM ON PM_DBA.SR_REMARK_SRRM
(ID_SR_R_PROJECT_SRRM)
LOGGING
NOPARALLEL;


CREATE INDEX PM_DBA.XIF2SR_REMARK_SRRM ON PM_DBA.SR_REMARK_SRRM
(ID_FROM_WHOM_SRRM)
LOGGING
NOPARALLEL;


CREATE INDEX PM_DBA.XIF3SR_REMARK_SRRM ON PM_DBA.SR_REMARK_SRRM
(ID_WHOM_SRRM)
LOGGING
NOPARALLEL;


CREATE INDEX PM_DBA.XIF4SR_REMARK_SRRM ON PM_DBA.SR_REMARK_SRRM
(ID_SR_R_PROMPTNESS_SRRM)
```

```
LOGGING
NOPARALLEL;
```

```
CREATE INDEX PM_DBA.XIF5SR_REMARK_SRRM ON PM_DBA.SR_REMARK_SRRM
(ID_SR_R_STATUS_SRRM)
LOGGING
NOPARALLEL;
```

```
CREATE INDEX PM_DBA."XIF6C3_3АМЕЧАНИЕ" ON PM_DBA.SR_REMARK_SRRM
(PARENT_REMARK_SRRM)
LOGGING
NOPARALLEL;
```

```
CREATE UNIQUE INDEX PM_DBA.XPKSR_REMARK_SRRM ON PM_DBA.SR_REMARK_SRRM
(ID_SR_REMARK_SRRM)
LOGGING
NOPARALLEL;
```

```
CREATE TABLE PM_DBA.SR_DETAIL_REMARK_SRDR
(
  ID_SR_DETAIL_REMARK_SRDR NUMBER(8) NOT NULL,
  ID_SR_REMARK_SRDR NUMBER(8) NOT NULL,
  NUMBER_PP_SRDR NUMBER(8) NOT NULL,
  DESCRIPTION_ACTION_SRDR VARCHAR2(2000 BYTE) NOT NULL,
  LINK_SRDR VARCHAR2(256 BYTE),
  VVOD_ID_CONTRACTOR_SRDR VARCHAR2(30 BYTE) NOT NULL,
  CHANGE_ID_CONTRACTOR_SRDR VARCHAR2(30 BYTE),
  DATE_INPUT_SRDR DATE DEFAULT sysdate NOT NULL,
  DATE_CHANGE_SRDR DATE
)
LOGGING
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;
```

```
COMMENT ON TABLE PM_DBA.SR_DETAIL_REMARK_SRDR IS 'Caption=СЗ_ДЕТАЛИЗАЦИЯ
ЗАМЕЧАНИЯ;Sequence=SEQ_SR_DETAIL_REMARK_SRDR';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.ID_SR_DETAIL_REMARK_SRDR IS
'ID_Детализация';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.ID_SR_REMARK_SRDR IS 'ID_Замечание';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.NUMBER_PP_SRDR IS 'Caption=№ п/п';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.DESCRPTION_ACTION_SRDR IS
'Описание последовательности действий';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.LINK_SRDR IS 'Ссылка на документ или
скриншот';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.VVOD_ID_CONTRACTOR_SRDR IS
'Автор ввода';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.CHANGE_ID_CONTRACTOR_SRDR IS  
'Автор изменения';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.DATE_INPUT_SRDR IS 'Дата ввода';
```

```
COMMENT ON COLUMN PM_DBA.SR_DETAIL_REMARK_SRDR.DATE_CHANGE_SRDR IS 'Дата  
изменения';
```

```
CREATE INDEX PM_DBA.XAK1SR_DETAIL_REMARK_SRDR ON PM_DBA.SR_DETAIL_REMARK_SRDR  
(ID_SR_REMARK_SRDR, NUMBER_PP_SRDR)  
LOGGING  
NOPARALLEL;
```

```
CREATE INDEX PM_DBA.XIF1SR_DETAIL_REMARK_SRDR ON PM_DBA.SR_DETAIL_REMARK_SRDR  
(ID_SR_REMARK_SRDR)  
LOGGING  
NOPARALLEL;
```

```
CREATE UNIQUE INDEX PM_DBA.XPKSR_DETAIL_REMARK_SRDR ON  
PM_DBA.SR_DETAIL_REMARK_SRDR  
(ID_SR_DETAIL_REMARK_SRDR)  
LOGGING  
NOPARALLEL;
```

```
ALTER TABLE PM_DBA.SR_R_CLIENT_SRCL ADD (  
CONSTRAINT XPKSR_R_CLIENT_SRCL  
PRIMARY KEY  
(ID_SR_R_CLIENT_SRCL)  
USING INDEX PM_DBA.XPKSR_R_CLIENT_SRCL  
ENABLE VALIDATE,  
CONSTRAINT XAK1SR_R_CLIENT_SRCL  
UNIQUE (NAME_CLIENT_SRCL)  
DEFERRABLE INITIALLY IMMEDIATE  
USING INDEX PM_DBA.XAK1SR_R_CLIENT_SRCL  
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_R_EXECUTIVE_SREX ADD (  
CONSTRAINT XPKSR_R_EXECUTIVE_SREX  
PRIMARY KEY  
(ID_SR_R_EXECUTIVE_SREX)  
USING INDEX PM_DBA.XPKSR_R_EXECUTIVE_SREX  
ENABLE VALIDATE,  
CONSTRAINT XAK1SR_R_EXECUTIVE_SREX  
UNIQUE (NUMBER_PP_SREX)  
DEFERRABLE INITIALLY IMMEDIATE  
USING INDEX PM_DBA.XAK1SR_R_EXECUTIVE_SREX  
ENABLE VALIDATE,  
CONSTRAINT XAK2SR_R_EXECUTIVE_SREX  
UNIQUE (LOGIN_SREX)  
DEFERRABLE INITIALLY IMMEDIATE  
USING INDEX PM_DBA.XAK2SR_R_EXECUTIVE_SREX  
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_R_PROMPTNESS_SRPR ADD (  
CONSTRAINT XPKSR_R_PROMPTNESS_SRPR
```

```
PRIMARY KEY
(ID_SR_R_PROMPTNESS_SRPR)
USING INDEX PM_DBA.XPKSR_R_PROMPTNESS_SRPR
ENABLE VALIDATE,
CONSTRAINT XAK1SR_R_PROMPTNESS_SRPR
UNIQUE (NUMBER_PP_SRPR)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XAK1SR_R_PROMPTNESS_SRPR
ENABLE VALIDATE,
CONSTRAINT XAK2SR_R_PROMPTNESS_SRPR
UNIQUE (NAME_SRPR)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XAK2SR_R_PROMPTNESS_SRPR
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_R_STATUS_SRST ADD (
CONSTRAINT XPKSR_R_STATUS_SRST
PRIMARY KEY
(ID_SR_R_STATUS_SRST)
USING INDEX PM_DBA.XPKSR_R_STATUS_SRST
ENABLE VALIDATE,
CONSTRAINT XAK1SR_R_STATUS_SRST
UNIQUE (NUMBER_PP_SRST)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XAK1SR_R_STATUS_SRST
ENABLE VALIDATE,
CONSTRAINT XAK2SR_R_STATUS_SRST
UNIQUE (NAME_STATUS_SRST)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XAK2SR_R_STATUS_SRST
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_PHYS_PERSON_SRPP ADD (
CONSTRAINT XPKSR_PHYS_PERSON_SRPP
PRIMARY KEY
(ID_SR_PHYS_PERSON_SRPP)
USING INDEX PM_DBA.XPKSR_PHYS_PERSON_SRPP
ENABLE VALIDATE,
CONSTRAINT XAK1SR_PHYS_PERSON_SRPP
UNIQUE (ID_SR_R_EXECUTIVE_SRPP)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XIF1SR_PHYS_PERSON_SRPP
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_R_PROJECT_SRPJ ADD (
CONSTRAINT XPKSR_R_PROJECT_SRPJ
PRIMARY KEY
(ID_SR_R_PROJECT_SRPJ)
USING INDEX PM_DBA.XPKSR_R_PROJECT_SRPJ
ENABLE VALIDATE,
CONSTRAINT XAK1SR_R_PROJECT_SRPJ
UNIQUE (NAME_PROJ_SRPJ)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XAK1SR_R_PROJECT_SRPJ
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_REMARK_SRRM ADD (
CONSTRAINT XPKSR_REMARK_SRRM
PRIMARY KEY
(ID_SR_REMARK_SRRM)
```

```
USING INDEX PM_DBA.XPKSR_REMARK_SRRM
ENABLE VALIDATE,
CONSTRAINT XAK1SR_REMARK_SRRM
UNIQUE (ID_SR_R_PROJECT_SRRM, NUMBER_PP_SRRM, DATE_CREATE_SRRM)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XAK1SR_REMARK_SRRM
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_DETAIL_REMARK_SRDR ADD (
CONSTRAINT XPKSR_DETAIL_REMARK_SRDR
PRIMARY KEY
(ID_SR_DETAIL_REMARK_SRDR)
USING INDEX PM_DBA.XPKSR_DETAIL_REMARK_SRDR
ENABLE VALIDATE,
CONSTRAINT XAK1SR_DETAIL_REMARK_SRDR
UNIQUE (ID_SR_REMARK_SRDR, NUMBER_PP_SRDR)
DEFERRABLE INITIALLY IMMEDIATE
USING INDEX PM_DBA.XAK1SR_DETAIL_REMARK_SRDR
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_PHYS_PERSON_SRPP ADD (
CONSTRAINT R_2849
FOREIGN KEY (ID_SR_R_EXECUTIVE_SRPP)
REFERENCES PM_DBA.SR_R_EXECUTIVE_SREX (ID_SR_R_EXECUTIVE_SREX)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_R_PROJECT_SRPJ ADD (
CONSTRAINT R_2848
FOREIGN KEY (ID_SR_PHYS_PERSON_SRPJ)
REFERENCES PM_DBA.SR_PHYS_PERSON_SRPP (ID_SR_PHYS_PERSON_SRPP)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE,
CONSTRAINT R_2850
FOREIGN KEY (ID_SR_R_CLIENT_SRPJ)
REFERENCES PM_DBA.SR_R_CLIENT_SRCL (ID_SR_R_CLIENT_SRCL)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_REMARK_SRRM ADD (
CONSTRAINT R_2835
FOREIGN KEY (ID_SR_R_PROJECT_SRRM)
REFERENCES PM_DBA.SR_R_PROJECT_SRPJ (ID_SR_R_PROJECT_SRPJ)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE,
CONSTRAINT R_2836
FOREIGN KEY (ID_FROM_WHOM_SRRM)
REFERENCES PM_DBA.SR_R_EXECUTIVE_SREX (ID_SR_R_EXECUTIVE_SREX)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE,
CONSTRAINT R_2837
FOREIGN KEY (ID_WHOM_SRRM)
REFERENCES PM_DBA.SR_R_EXECUTIVE_SREX (ID_SR_R_EXECUTIVE_SREX)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE,
CONSTRAINT R_2838
FOREIGN KEY (ID_SR_R_PROMPTNESS_SRRM)
REFERENCES PM_DBA.SR_R_PROMPTNESS_SRPR (ID_SR_R_PROMPTNESS_SRPR)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE,
```

```
CONSTRAINT R_2839
FOREIGN KEY (ID_SR_R_STATUS_SRRM)
REFERENCES PM_DBA.SR_R_STATUS_SRST (ID_SR_R_STATUS_SRST)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE,
CONSTRAINT R_2841
FOREIGN KEY (PARENT_REMARK_SRRM)
REFERENCES PM_DBA.SR_REMARK_SRRM (ID_SR_REMARK_SRRM)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE);
```

```
ALTER TABLE PM_DBA.SR_DETAIL_REMARK_SRDR ADD (
CONSTRAINT R_2840
FOREIGN KEY (ID_SR_REMARK_SRDR)
REFERENCES PM_DBA.SR_REMARK_SRRM (ID_SR_REMARK_SRRM)
DEFERRABLE INITIALLY IMMEDIATE
ENABLE VALIDATE);
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON
PM_DBA.SR_R_CLIENT_SRCL TO CUSTOM;
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON
PM_DBA.SR_R_EXECUTIVE_SREX TO CUSTOM;
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON
PM_DBA.SR_R_PROMPTNESS_SRPR TO CUSTOM;
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON
PM_DBA.SR_R_STATUS_SRST TO CUSTOM;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_R_CLIENT_SRCL TO PUBLIC;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_R_EXECUTIVE_SREX TO PUBLIC;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_R_PROMPTNESS_SRPR TO PUBLIC;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_R_STATUS_SRST TO PUBLIC;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_R_CLIENT_SRCL TO RTIME_DBA;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_R_EXECUTIVE_SREX TO RTIME_DBA;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_R_PROMPTNESS_SRPR TO RTIME_DBA;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_R_STATUS_SRST TO RTIME_DBA;
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON
PM_DBA.SR_PHYS_PERSON_SRPP TO CUSTOM;
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON
PM_DBA.SR_R_PROJECT_SRPJ TO CUSTOM;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_PHYS_PERSON_SRPP TO PUBLIC;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_R_PROJECT_SRPJ TO PUBLIC;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,  
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_PHYS_PERSON_SRPP TO RTIME_DBA;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,  
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_R_PROJECT_SRPJ TO RTIME_DBA;
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON  
PM_DBA.SR_REMARK_SRRM TO CUSTOM;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_REMARK_SRRM TO PUBLIC;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,  
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_REMARK_SRRM TO RTIME_DBA;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,  
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_REMARK_SRRM TO S_DBA;
```

```
GRANT ALTER, INDEX, REFERENCES, ON COMMIT REFRESH, QUERY REWRITE, FLASHBACK ON  
PM_DBA.SR_DETAIL_REMARK_SRDR TO CUSTOM;
```

```
GRANT DELETE, INSERT, SELECT, UPDATE ON PM_DBA.SR_DETAIL_REMARK_SRDR TO PUBLIC;
```

```
GRANT ALTER, DELETE, INDEX, INSERT, REFERENCES, SELECT, UPDATE, ON COMMIT REFRESH,  
QUERY REWRITE, DEBUG, FLASHBACK ON PM_DBA.SR_DETAIL_REMARK_SRDR TO RTIME_DBA;
```