

Руководство администратора
Программный комплекс «ТМ:ЦОД.МЭК»

На 123 листах

2020 г.

Содержание

1 ОБЩИЕ СВЕДЕНИЯ	7
1.1 НАЗНАЧЕНИЕ ПРОГРАММЫ	7
1.2 СВЕДЕНИЯ О ТЕХНИЧЕСКИХ СРЕДСТВАХ	7
1.3 СВЕДЕНИЯ О ПРОГРАММНЫХ СРЕДСТВАХ	7
1.4 ОСНОВНЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	8
2 НАСТРОЙКА «ТМ:ЦОД.МЭК»	9
2.1 АРМ «АДМИНИСТРИРОВАНИЕ»	10
2.1.1 <i>Подключения</i>	10
2.1.1.1 Добавление подключения	11
2.1.1.2 Редактирование подключения	13
2.1.1.3 Удаление подключения	13
2.1.1.4 Клонировать подключение	13
2.1.1.5 Запоминать логин	13
2.1.2 <i>Пользователи</i>	13
2.1.2.1 Добавление пользователя	14
2.1.2.2 Редактирование пользователя	15
2.1.2.3 Удаление пользователя	16
2.1.3 <i>Группы пользователей</i>	16
2.1.3.1 Добавление группы пользователей	17
2.1.3.2 Редактирование группы пользователей	19
2.1.3.3 Удаление группы пользователей	20
2.1.4 <i>Роли</i>	20
2.1.4.1 Добавление роли	21
2.1.4.2 Редактирование роли	23
2.1.4.3 Удаление роли	23
2.1.5 <i>Группы документов</i>	24
2.1.5.1 Добавление группы	25
2.1.5.2 Редактирование группы	25
2.1.5.3 Удаление группы	25
2.1.6 <i>Библиотеки</i>	26
2.1.6.1 Подключение библиотек	26
2.1.6.2 Редактирование записи библиотеки	27
2.1.6.3 Отключение библиотек	27
2.1.7 <i>Настройки</i>	28
2.1.7.1 Добавление пользовательских настроек	29
2.1.7.2 Редактирование пользовательских настроек	31
2.1.7.3 Удаление пользовательских настроек	31
2.1.8 <i>Документы</i>	32

2.1.8.1	Настройка индексов.....	32
2.1.9	<i>Отчеты</i>	35
2.1.9.1	Добавление отчетов	36
2.1.9.2	Редактирование отчетов.....	37
2.1.9.3	Удаление отчетов.....	37
2.1.10	<i>Протоколы</i>	38
2.1.10.1	Добавление протоколов	38
2.1.10.2	Редактирование протоколов.....	39
2.1.10.3	Удаление протоколов.....	39
2.1.11	<i>Права доступа</i>	40
2.1.11.1	Права по умолчанию	40
2.1.11.2	Заданные права.....	40
2.1.11.3	Результирующие права	41
2.2	ПОДПУНКТЫ МЕНЮ «ОПЕРАЦИИ»	41
2.2.1	<i>Диагностика</i>	41
2.2.1.1	Переиндексация таблиц	41
2.2.1.2	Управление логическими блокировками	43
2.2.1.3	Проверка ссылочной целостности	44
2.2.2	<i>Журналирование</i>	45
2.2.2.1	Очистка журналов.....	46
2.2.2.2	Журнал системы	47
2.2.2.3	Журнал обновлений	48
2.2.2.4	Журнал протоколов	49
2.2.2.5	Журнал отчетов.....	50
2.2.3	<i>Резервное копирование базы данных</i>	52
2.2.3.1	Создание резервной копии базы данных.....	52
2.2.3.2	Восстановление базы данных из резервной копии	53
2.2.4	<i>Настройка приложения</i>	53
2.2.5	<i>Реструктуризация базы данных</i>	54
2.2.5.1	Настройка формирования отчетов	55
2.3	МЕТОДЫ И СЦЕНАРИИ УСТАНОВКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАБОЧИХ СТАНЦИЙ И СЕРВЕРОВ	55
2.4	МЕТОДЫ ПРОВЕРКИ РАБОТСПОСОБНОСТИ РАБОЧИХ СТАНЦИЙ И СЕРВЕРОВ.....	56
2.5	МЕТОДЫ ВОССТАНОВЛЕНИЯ РАБОТСПОСОБНОСТИ РАБОЧИХ СТАНЦИЙ И СЕРВЕРОВ.....	60
2.6	МЕТОДЫ ПРОВЕРКИ РАБОТСПОСОБНОСТИ БД.....	60
2.7	МЕТОДЫ ПОДДЕРЖАНИЯ ЦЕЛОСТНОСТИ БД	60
2.8	МЕТОДЫ ПОДДЕРЖАНИЯ БЕЗОПАСНОСТИ БД.....	60
2.9	МЕТОДЫ ВОССТАНОВЛЕНИЯ РАБОТСПОСОБНОСТИ БД.....	69
ПРИЛОЖЕНИЕ А ПОДСИСТЕМА ОТЧЕТОВ		70

А.1. СИСТЕМА ОПИСАНИЯ ОТЧЕТОВ АДМИНИСТРАТОРОМ. ФОРМА РЕДАКТИРОВАНИЯ ОТЧЕТОВ	70
.....	70
А.1.1. Свойства отчета	71
А.1.1.1. Типы отчетов и их особенности	71
А.1.1.2. Форматы представления отчетов	72
А.1.2. Параметры отчетов	73
А.1.2.1. Интерактивные параметры	73
А.1.2.2. Доступ к значениям экземпляра документа и набора документов в отчетах	74
А.1.2.3. Константы отчета – параметры на основе пользовательских настроек	74
А.1.2.4. Доступ к значениям параметров. Особое значение WasInited	75
А.1.2.5. Специальные параметры (зарезервированные имена)	75
А.1.3. Таблицы данных отчетов	76
А.1.3.1. Запросы для получения таблиц данных	76
А.1.3.2. Правила задания подстановок значений параметров в текст запроса	77
А.1.3.3. Группировка данных таблицы с подсчетом промежуточных итогов	77
А.1.4. Тестирование отчетов. Выгрузка/загрузка описателей отчетов.	77
А.2. ПРАВИЛА СОЗДАНИЯ EXCEL-ШАБЛОНОВ ОТЧЕТОВ	78
А.2.1. Общие рекомендации	78
А.2.2. Подстановка значений параметров (&&Param.)	78
А.2.2.1. Подстановка значений параметров в колоннотитулы (@@Param.)	79
А.2.3. Рекомендации по оформлению шапки таблицы	79
А.2.4. Описатель начала подстановки данных таблицы отчета (&&Tab.)	81
А.2.5. Порядковые номера записей в таблице (.RecordNumber).	81
А.2.6. Диапазон подстановки полей (&&Details)	81
А.2.7. Для чего и как нужно применять &&TabEnd	82
А.2.8. Описатели, используемые для задания сводных таблиц	82
А.2.8.1. Разметка региона – источника сводной таблицы (&&PivotTableBegin и &&PivotTableEnd)	82
А.2.8.2. Строки, колонки и данные сводных таблиц	82
А.2.8.3. Приемник сводной таблицы	83
А.2.8.4. Построение диаграммы на основе сводной таблицы	83
А.2.9. Пост-макросы в Excel-отчетах	83
ПРИЛОЖЕНИЕ Б ПОДСИСТЕМА DTS	85
Б.1. НАЗНАЧЕНИЕ ПОДСИСТЕМЫ ЭКСПОРТА-ИМПОРТА	85
Б.2. ПРОТОКОЛ	85
Б.2.1. Управление протоколами	85
Б.2.2. Структура протокола	87
Б.2.3. Выполнение протокола	88
Б.3 ОСНОВНЫЕ СВОЙСТВА ПРОТОКОЛА	88

<i>Б.3.1. Общие свойства</i>	<i>88</i>
<i>Б.3.2. Привязка к документу</i>	<i>90</i>
<i>Б.3.3. Поведенческие свойства.....</i>	<i>90</i>
<i>Б.3.4. Режим отладки.....</i>	<i>91</i>
<i>Б.3.5. История редактирования.....</i>	<i>91</i>
Б.4. ИСПОЛНЯЕМЫЕ ЭЛЕМЕНТЫ ПРОТОКОЛА	92
<i>Б.4.1. Параметры</i>	<i>94</i>
<i>Б.4.1.1. Редактор параметров.....</i>	<i>94</i>
<i>Б.4.1.2. Использование параметров</i>	<i>96</i>
<i>Б.4.2.3. Служебные параметры</i>	<i>97</i>
<i>Б.4.2. Скрипты</i>	<i>98</i>
<i>Б.4.3. Отчеты.....</i>	<i>98</i>
<i>Б.4.4. Источник MSSQL</i>	<i>102</i>
<i>Б.4.5. Источник XML.....</i>	<i>103</i>
<i>Б.4.6. Источник DBF</i>	<i>109</i>
<i>Б.4.7. Приемник MSSQL</i>	<i>111</i>
<i>Б.4.8. Приемник XML.....</i>	<i>113</i>
<i>Б.4.9. Приемник DBF</i>	<i>121</i>

ПРИНЯТЫЕ СОКРАЩЕНИЯ

Т – SQL	Расширение языка SQL компаний Microsoft (для Microsoft SQL Server) и Sybase (для Sybase ACE)
БД	База данных
ИС	Информационная система
ЛПУ	Лечебно-профилактическое учреждение
НСИ	Нормативно-справочная информация
ОМС	Обязательное медицинское страхование
ОС	Операционная система
ФК	Функциональная компонента
ПО	Программное обеспечение
Протокол преобразования данных, протокол импорта/экспорта	xml – файл с набором правил преобразования данных. В качестве формата описания данных используются xsd-схемы. Преобразования и контроль данных описываются с использованием языка Т – SQL.
Реструктуризация БД	Изменение структуры БД средствами модуля администрирования, путем выполнения пакета реструктуризации
РСО, Реестр счетов	Реестров счетов для оплаты расходов по проведению дополнительной диспансеризации граждан РФ по законченному случаю дополнительной диспансеризации
РФ	Российская Федерация
Смежные информационные системы	ИС, функционирующая на базе одного из объектов автоматизации и с которой регламентирован информационный обмен
СМО	Страховая медицинская организация
СУБД	Система управления БД
ТЗ	Техническое задание
ТП	Техническая поддержка
ТФОМС	Территориальный фонд обязательного медицинского страхования
ФОМС	Федеральный фонд обязательного медицинского страхования
ЭД	Эксплуатационная документация

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение программы

Полное наименование комплекса: Региональный центр обработки данных. Медико-экономический контроль.

Сокращённое обозначение комплекса: «ТМ:ЦОД.МЭК».

«ТМ:ЦОД.МЭК» предназначен для персонифицированного учета оказанных медицинских услуг, распространения федеральных и ведения региональных сегментов нормативно-справочной информации и организации информационного взаимодействия участников в сфере ОМС.

Настоящее руководство администратора содержит подробные инструкции по настройке и дальнейшему сопровождению программного комплекса «ТМ:ЦОД.МЭК».

1.2 Сведения о технических средствах

Для корректного функционирования Программы необходима следующая (минимальная) конфигурация технических средств:

Наименование	Сервер тип 1 (более 100 раб.мест)	Сервер тип 2 (до 50 раб.мест)	Сервер тип 3 (до 20 раб.мест)	Клиент - рабочая станция (десктоп)
Процессор	Кол-во ядер: не менее 24	Кол-во ядер: не менее 12	Кол-во ядер: не менее 8	Кол-во ядер: не менее 2
Оперативная память	Не менее 64GB	Не менее 32GB	Не менее 16GB	Не менее 2GB
Дисковый массив	Не менее 3Tb	Не менее 1Tb	Не менее 1Tb	Не менее 60Gb
Сетевая карта	Gigabit Ethernet Controller	Gigabit Ethernet Controller	Gigabit Ethernet Controller	Gigabit Ethernet Controller

1.3 Сведения о программных средствах

Для функционирования Программы необходимо следующее программное обеспечение:

Для сервера:

- Операционная система семейства MS Windows Server версии не ниже 2012 R2;
- Наличие MS .NET FrameWork 3.5 SP1;
- Наличие MS .NET FrameWork 4.0;
- Наличие SQL-сервера семейства MS SQL Server версии не ниже 2014 Standart Edition или выше, включая Analysis Services;
- Наличие MS Office 2010/2013 или OpenOffice Calc версии 3.3.0;

Для рабочей станции (десктоп):

- Операционная система семейства MS Windows не ниже XP Prof SP3;
- Наличие MS Office 2010/2013 или OpenOffice Calc версии 3.3.0;
- Браузер IE 9+, или Chrome 10+, или FireFox 4+.

1.4 Основные термины и определения

Подключения – наименование и информация, необходимая для подключения к базе данных – имя сервера, имя базы данных, логин и пароль.

Классы – документы, доступные в «ТМ:ЦОД.МЭК». Каждый класс соответствует отдельному документу, доступному в «ТМ:ЦОД.МЭК».

Представления – сокращенный вид документа (сокращенный набор полей). Используется для ускорения работы системы.

Группы – объединение (группировка) нескольких классов. Группы определяют тематические разделы, которые будут отображаться в программе.

Роли – набор прав доступа для пользователей. Роли определяют, какие классы и группы будут доступны пользователю и тип доступа к классам и к группам. Также роли определяют, какие библиотеки, представления и операции будут доступны пользователем, для которых будет установлена данная роль.

Пользователи – пользователи программного комплекса. Каждый пользователь может выполнять одну или несколько назначенных ему ролей. Для каждого подключения должен существовать, по крайней мере, один пользователь.

Библиотеки – dll-библиотеки, содержащие набор дополнительных функций для системы. Библиотеки нужны для определения конкретного интерфейса «ТМ:ЦОД.МЭК».

Операция – последовательность запрограммированных действий «ТМ:ЦОД.МЭК» в данной предметной области.

Пользовательские настройки – пользовательские настройки, определенные в «ТМ:ЦОД.МЭК».

2 НАСТРОЙКА «ТМ:ЦОД.МЭЖ»

Основные элементы и операции Администратора располагаются в окне АРМа «Администрирование», а также в подпунктах главного меню «Операции» (Рисунок 1).

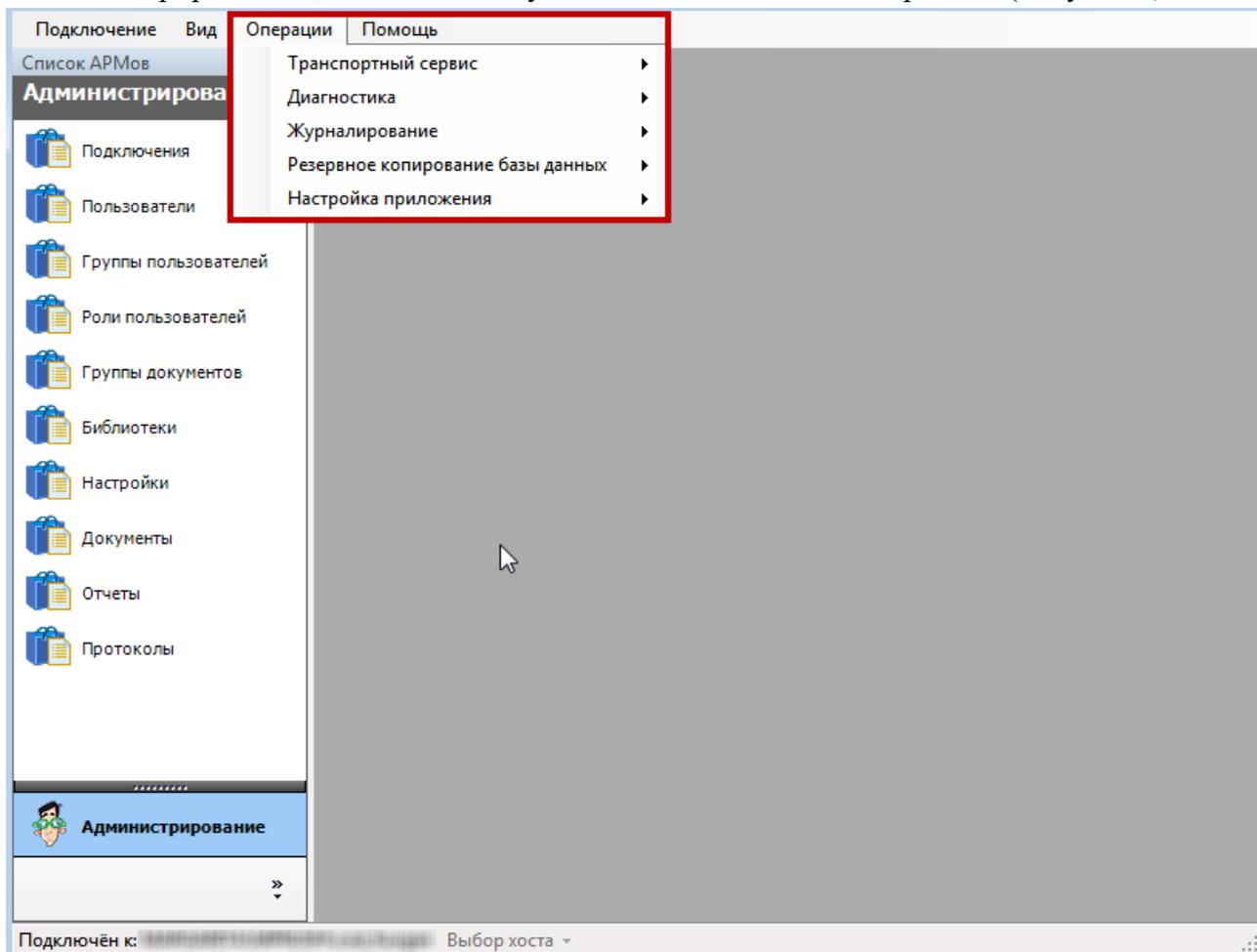


Рисунок 1. Основные элементы интерфейса Программы в режиме администрирования

Краткий перечень операций и элементов администрирования приведен в таблицах (Таблица 1, Таблица 2). Более подробное описание операций и элементов администрирования приведено в соответствующих разделах документа.

Таблица 1. Перечень элементов АРМа «Администрирование»

Имя элемента	Назначение	Примечание
Подключения	Управление подключениями к базам данных.	Схоже с настройкой подключений доступной из окна авторизации. Включено в администраторе с минимальным набором операций.
Пользователи	Управление составом пользователей	
Роли	Управление составом ролей	
Группы	Управление составом групп документов	
Библиотеки	Управление подключаемыми библиотеками	
Настройки	Управление пользовательскими настройками	
Отчеты	Управление отчетами	
Протоколы	Управление протоколами	

	импорта/экспорта	
Документы	Просмотр типов документов, зарегистрированных в системе	Так же позволяет выполнить настройку индексов и переиндексацию в БД.

Таблица 2. Перечень операций администрирования

Имя подпункта меню «Операции»	Назначение	Примечание
Диагностика. Переиндексация таблиц	Диалог управления переиндексацией таблиц.	
Диагностика. Управление логическими блокировками	Диалог управления заблокированными документами.	
Диагностика. Проверка ссылочной целостности	Диалог контроля целостности данных в БД.	
Журналирование. Очистка журналов	Очистка журналов системы, протоколов, отчетов, документов	
Журналирование. Журнал системы	Журнал просмотра истории операций выполненных системой	
Журналирование. Журнал протоколов	Журнал просмотра истории выполненных протоколов	
Журналирование. Журнал отчетов	Журнал просмотра истории выполненных отчетов	
Резервное копирование базы данных. Создание резервной копии	Операция сохранения копии БД.	
Резервное копирование базы данных. Восстановление резервной копии	Операция восстановления копии БД.	Включено в администраторе с минимальным набором операций.
Настройка приложения. Реструктуризация базы данных	Выполнить операцию регламентного переноса структуры БД.	
Настройка приложения. Настройка формирования отчетов	Диалог выбора приложения для формирования отчетов.	

2.1 АРМ «Администрирование»

2.1.1 Подключения

Подключение – служебная запись, содержащая информацию для подключения приложения к БД. Все записи с подключениями хранятся зашифрованными в служебном файле `connections.config`, находящемся в рабочей папке приложения.

Следует обратить внимание на тот факт, что подключение и данные авторизации, указанные в нем, используются для соединения с БД. Что же касается логина и пароля, вводимых в окне авторизации, они используются для авторизации непосредственно в самом приложении. Т.е. логин с паролем, указываемые в подключении, и логин с паролем, вводимые при входе пользователя, различны.

Список доступных подключений можно увидеть, как в АРМ «Администрирование» на вкладке «Подключения», так и непосредственно в окне авторизации. Управление подключениями также может осуществляться в двух местах:

1. На вкладке «Подключения» в приложении администрирования.

2. Через встроенный редактор в окне авторизации. Редактор вызывается кнопкой , расположенной в левом нижнем углу окна авторизации.

И в том, и в другом случае пользователю предоставляется список подключений и панель кнопок для управления ими (Рисунок 2).

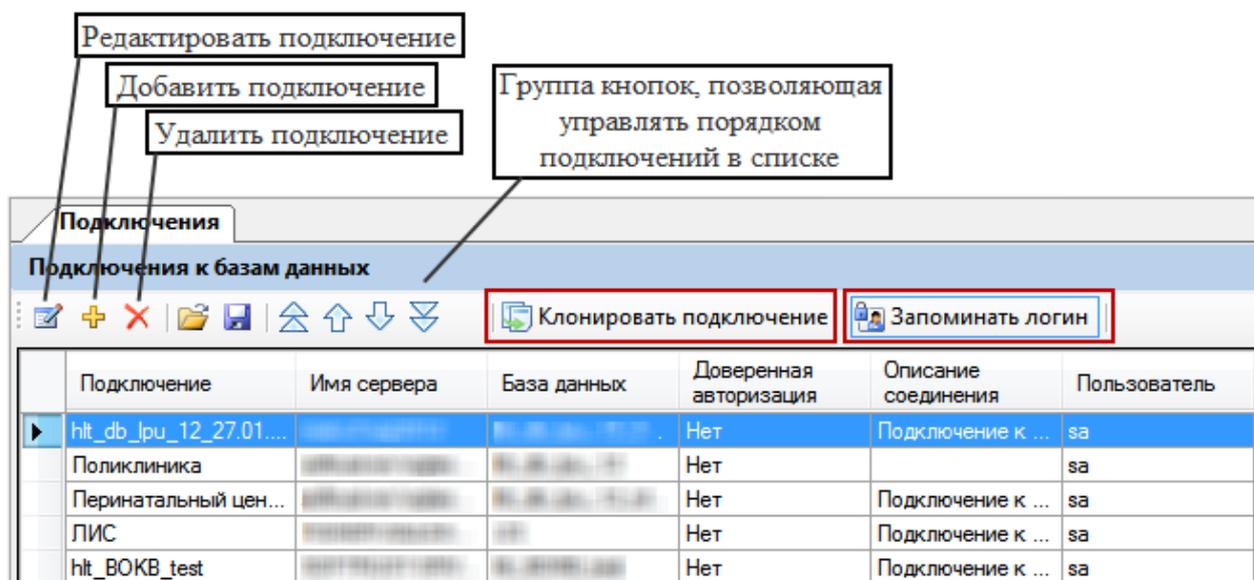


Рисунок 2. Панель управления подключениями

2.1.1.1 Добавление подключения

Для добавления подключения необходимо воспользоваться соответствующей кнопкой. Будет открыт диалог для создания нового подключения (Рисунок 3).

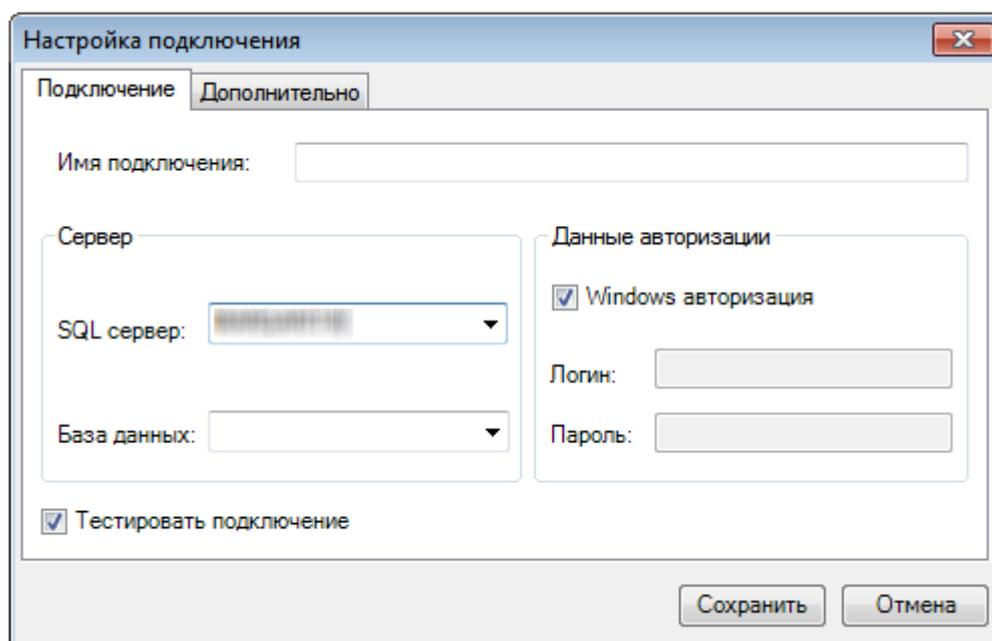


Рисунок 3. Окно создания нового подключения

Рассмотрим поля ввода, которые необходимо заполнить:

- «Имя подключения» – название создаваемого подключения. Данное наименование будет отображаться в форме авторизации.
- «SQL сервер» – имя компьютера в сети, на котором развернут SQL-сервер, а также, если существует, имя SQL-сервера, к которому будет осуществляться подключение. В качестве разделителя имени компьютера и имени сервера используется символ «\». Если SQL-сервер неименованный, то указывается только имя компьютера.

- «База данных» – имя базы данных, к которой будет производиться подключение. База данных должна существовать на указанном сервере, иначе при попытке подключения возникнет ошибка.
- Флаг «Windows-авторизация» – если он установлен, при подключении к SQL-серверу будет использована информация о текущем пользователе Windows. В этом случае заполнение полей «Пользователь» и «Пароль» не требуется.
- «Логин»– имя пользователя для подключения к SQL-серверу.
- «Пароль» – пароль для подключения к SQL-серверу.
- Флаг «Тестировать подключение» – если установлен, то при сохранении подключения будет произведена попытка установить соединение. Если соединение будет успешно установлено, оно будет сохранено. Если нет – пользователю будет предложено исправить настройки или сохранить параметры, не прошедшие проверку.

Вкладка «Дополнительно» содержит ряд технических параметров подключения (Рисунок 4). Параметры с этой вкладки необходимо изменять только в том случае, когда существуют проблемы с подключением к SQL-серверу.

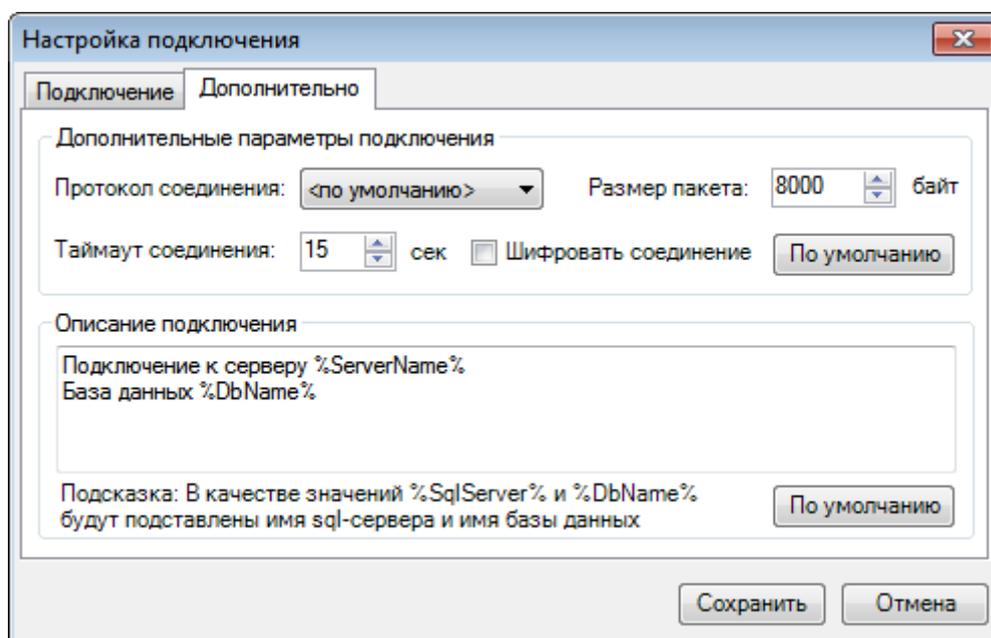


Рисунок 4. Дополнительные параметры подключения

Доступные для управления параметры:

- Протокол соединения – один из возможных вариантов протоколов для соединения с SQL-сервером.
- Размер пакета – размер пакета данных в байтах.
- Таймаут соединения – время ожидания в секундах, в течение которого соединение должно быть установлено (по умолчанию 15 секунд).
- Описание подключения – описание подключения с использованием предопределённых имён: %ServerName% – имя SQL-сервера, %DbName% – имя базы данных.

Восстановить значения умолчания можно с помощью кнопок «По умолчанию».

После ввода необходимых данных для сохранения подключения необходимо нажать кнопку «Сохранить». Если подключение не требуется сохранять – нажать кнопку «Отмена».

2.1.1.2 Редактирование подключения

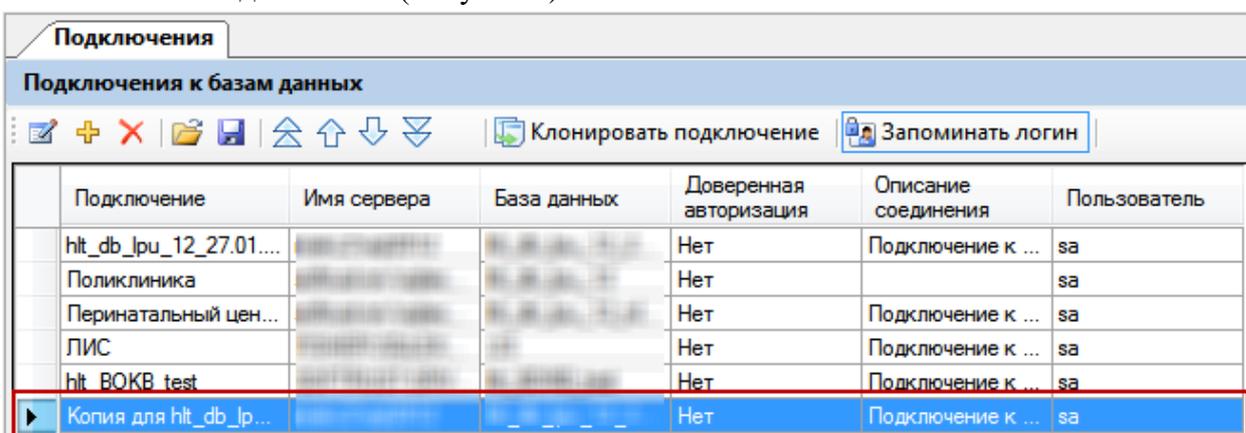
Для редактирования подключения необходимо воспользоваться соответствующей кнопкой. Открывшийся диалог будет полностью аналогичен диалогу для создания подключения (Рисунок 3). В нем можно изменить наименование подключения, сервер, базу данных и прочие параметры. Для сохранения изменения необходимо нажать кнопку «Сохранить», для отмены – кнопку «Отмена».

2.1.1.3 Удаление подключения

Для удаления подключения необходимо выбрать соответствующее подключение в списке, и нажать кнопку «Удалить».

2.1.1.4 Клонировать подключение

Для копирования подключения необходимо выбрать подключение из списка и нажать кнопку  Клонировать подключение. В созданной записи нужно откорректировать наименование подключения (Рисунок 5).



Подключение	Имя сервера	База данных	Доверенная авторизация	Описание соединения	Пользователь
hlt_db_ipu_12_27.01....			Нет	Подключение к ...	sa
Поликлиника			Нет		sa
Перинатальный цен...			Нет	Подключение к ...	sa
ЛИС			Нет	Подключение к ...	sa
hlt ВОКВ test			Нет	Подключение к ...	sa
Копия для hlt_db_ip...			Нет	Подключение к ...	sa

Рисунок 5. Клонирование подключения

2.1.1.5 Запоминать логин

Для удобной работы при переподключении пользователя можно сохранять логин последнего подключения. Для этого нужно нажать кнопку  Запоминать логин (Рисунок 2).

2.1.2 Пользователи

На вкладке «Пользователи» отображается список всех зарегистрированных в программе пользователей (Рисунок 6). На данной вкладке можно добавлять учетные записи новых пользователей, редактировать уже существующие и удалять ненужные.

Учетная запись пользователя «Admin» является служебной. Удаление данной записи запрещено. Настройка ролей в данной учетной записи недоступна. Суперпользователь «Admin» всегда обладает полными правами на все элементы программы. Редактирование (в том числе и переименование) служебной записи «Admin» разрешено.

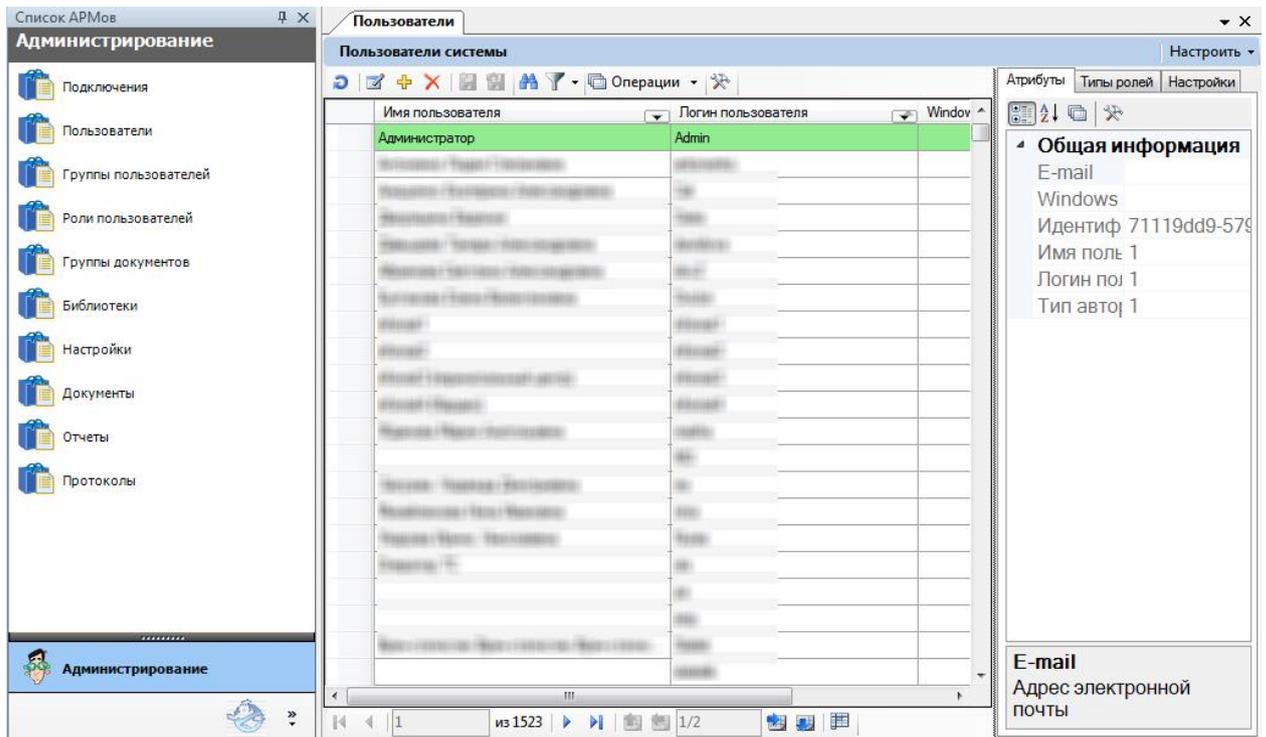


Рисунок 6. Окно управления пользователями

2.1.2.1 Добавление пользователя

Для добавления новой учетной записи необходимо нажать соответствующую кнопку на панели инструментов или выбрать пункт «Добавить» в контекстном меню. В открывшемся диалоге «Добавление учетной записи пользователя» (Рисунок 7) содержатся закладки «Пользователь» и «Роли пользователя».

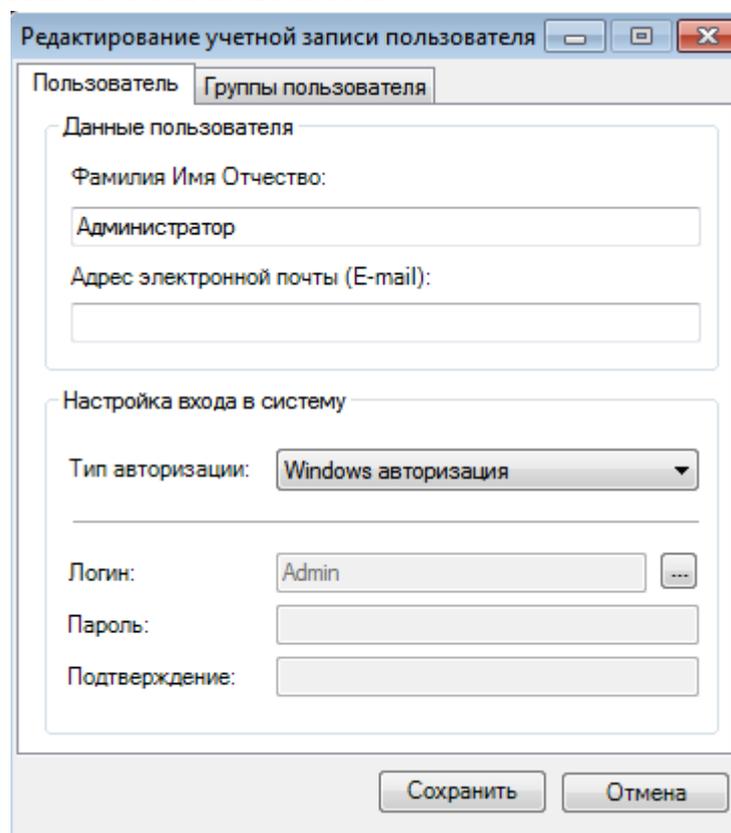


Рисунок 7. Диалог создания новой учетной записи пользователя

Описание полей на вкладке «Пользователь»:

- Фамилия Имя Отчество – фамилия, имя и отчество пользователя.
- Адрес электронной почты (E-mail).
- Тип авторизации – выпадающий список, определяющий, какая авторизация разрешена пользователю для входа в программу. В зависимости от типа авторизации может потребоваться заполнить все поля или только часть. Тип авторизации может принимать значения:

1. Общая авторизация – для входа в программу необходимо указывать логин и пароль.
2. Windows авторизация – для входа в программу будет использовано имя пользователя из операционной системы Windows.
3. Смешанная авторизация – для входа в программу пользователя может использоваться любой из видов авторизации.

- Windows-логин – имя пользователя сеанса операционной системы Windows.
- Логин – учетное имя пользователя, под которым пользователь выполняет вход в систему.

- Пароль – пароль, с которым пользователь выполняет вход в программу.
- Подтверждение – подтверждение пароля, с которым пользователь выполняет вход в программу.

Вкладка «Группы пользователя» (Рисунок 8) позволяет включить пользователя в определенную группу пользователей.

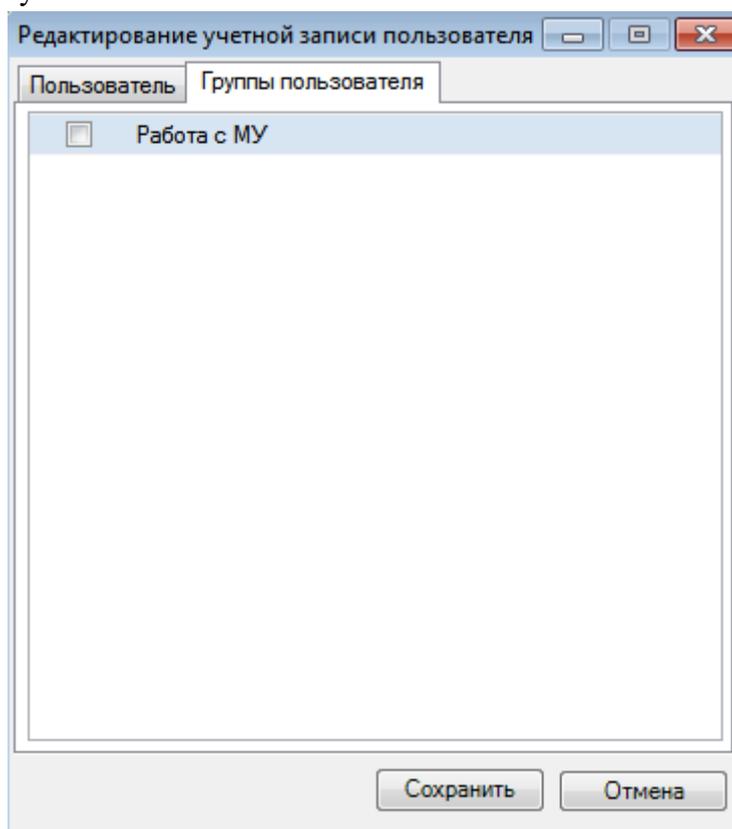


Рисунок 8. Управление группами пользователя

2.1.2.2 Редактирование пользователя

Для редактирования данных пользователя необходимо выбрать пользователя в списке, и нажать соответствующую кнопку на панели инструментов, либо выбрать пункт

«Редактировать» в контекстном меню. Полученный диалог «Редактирование пользователя» полностью аналогичен диалогу «Добавление пользователя» (Рисунок 7). В нем можно изменить учетное имя пользователя, его ФИО, тип авторизации, доменное имя компьютера, а также роли, доступные данному пользователю. Чтобы сохранить внесенные изменения, необходимо нажать кнопку «Сохранить»; для отмены – кнопку «Отмена».

2.1.2.3 Удаление пользователя

Для удаления пользователя нужно выбрать его в списке пользователей и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 9) запись будет удалена. Также можно удалить несколько учетных записей, выбрав их в списке.

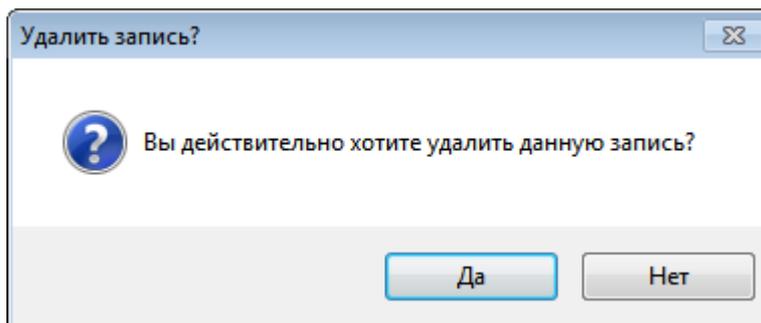


Рисунок 9. Диалог подтверждения удаления

2.1.3 Группы пользователей

На вкладке «Группы пользователей» отображается список всех определенных в программе групп пользователей (Рисунок 10). Список групп пользователя также можно просмотреть в форме редактирования учетных записей на вкладке «Группы пользователей» при просмотре списка пользователей программы (Рисунок 8).

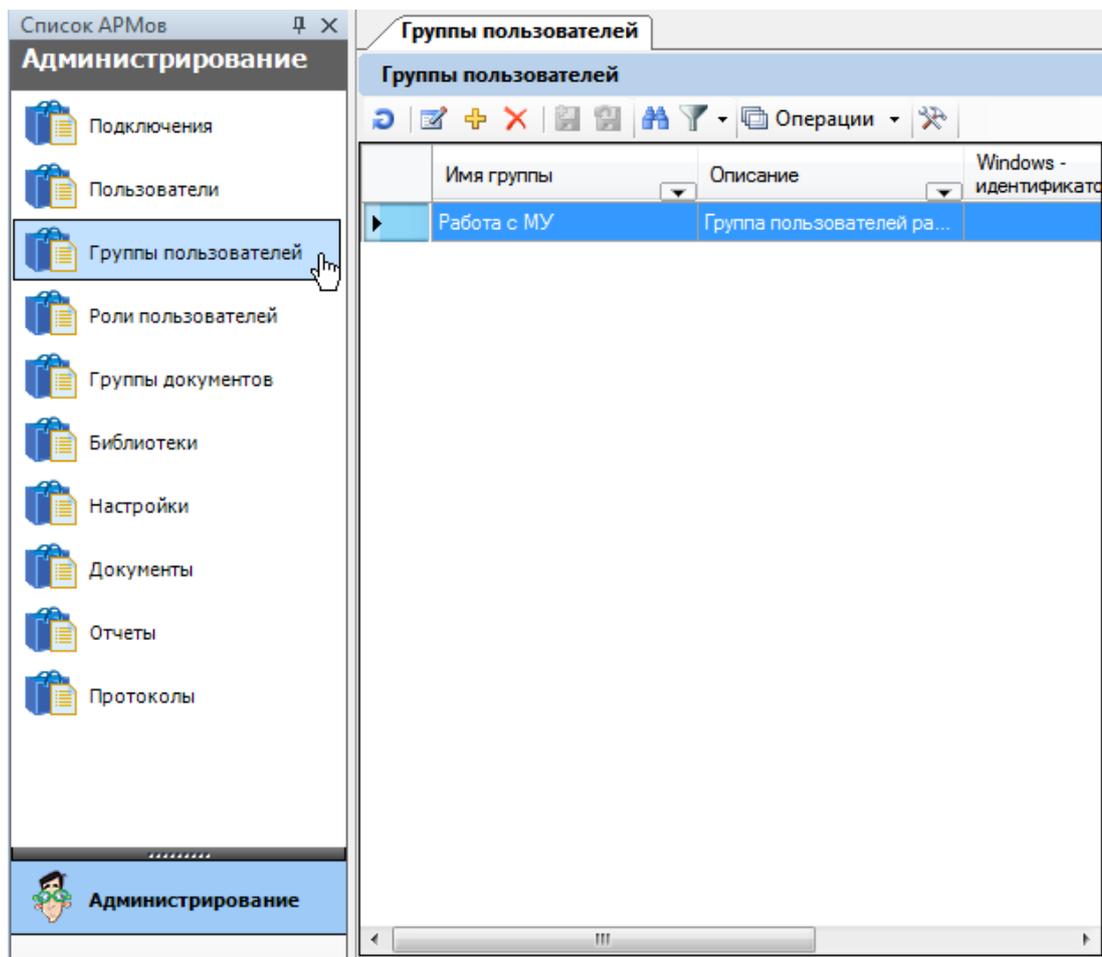


Рисунок 10. Список зарегистрированных ролей

2.1.3.1 Добавление группы пользователей

Добавление группы пользователей выполняется с помощью соответствующего диалога (Рисунок 11).

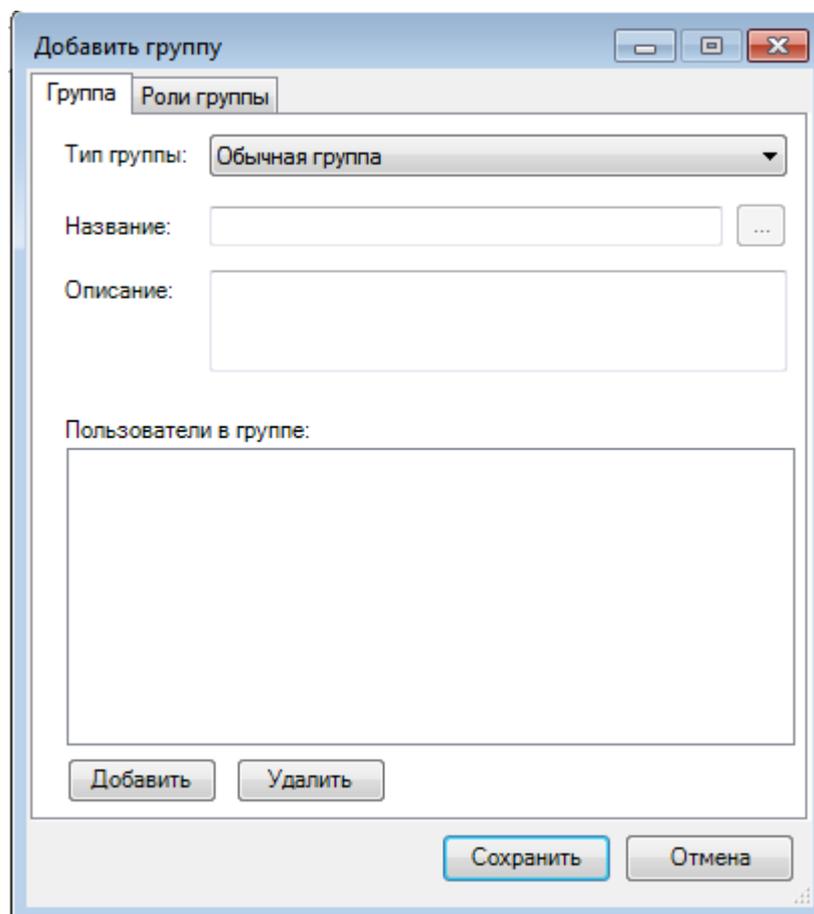


Рисунок 11. Диалог «Добавить группу», вкладка «Группа»

При добавлении группы пользователей на вкладке «Группа» необходимо заполнить два обязательных поля:

- Название – имя, под которым группа пользователей фигурирует в программе.
- Тип группы – атрибут, определяющий, на какие элементы системы распространяется данная группа. Может принимать значения «Обычная группа» и «Windows группа».

Для добавления пользователей в группу нужно нажать на кнопку «Добавить». Для удаления пользователя нужно выделить нужную запись и нажать на кнопку «Удалить».

На вкладке «Роли группы» необходимо проставить галочку около требуемых ролей группы пользователей (Рисунок 12).

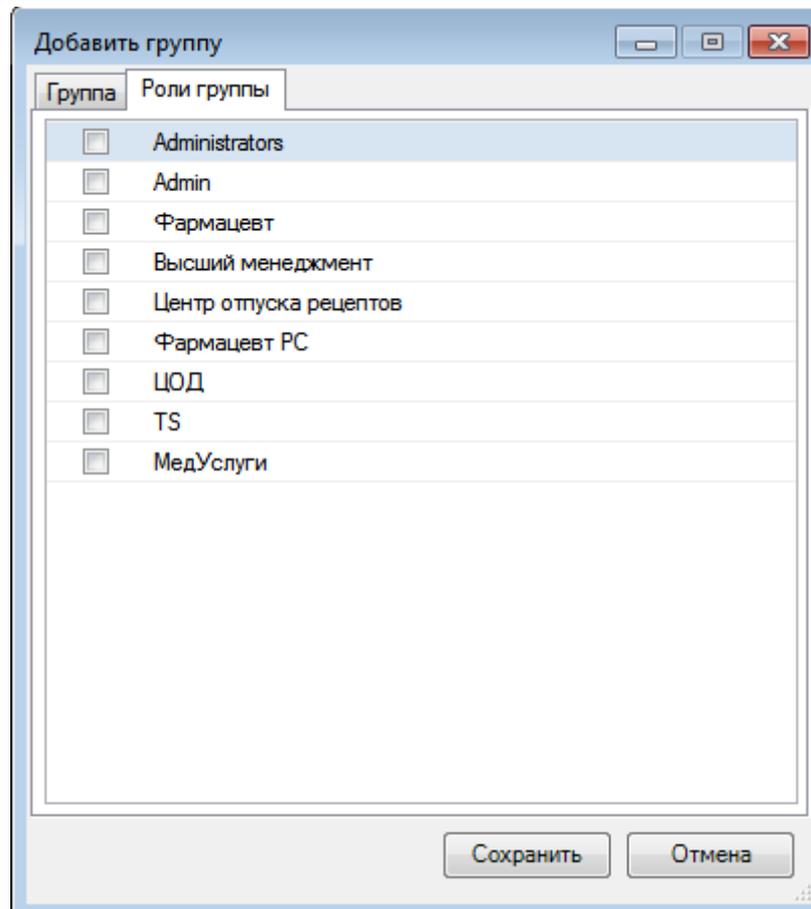


Рисунок 12. Диалог «Добавить группу», вкладка «Роли группы»

2.1.3.2 Редактирование группы пользователей

Редактирование группы пользователей выполняется в том же диалоговом окне, что и добавление (Рисунок 13).

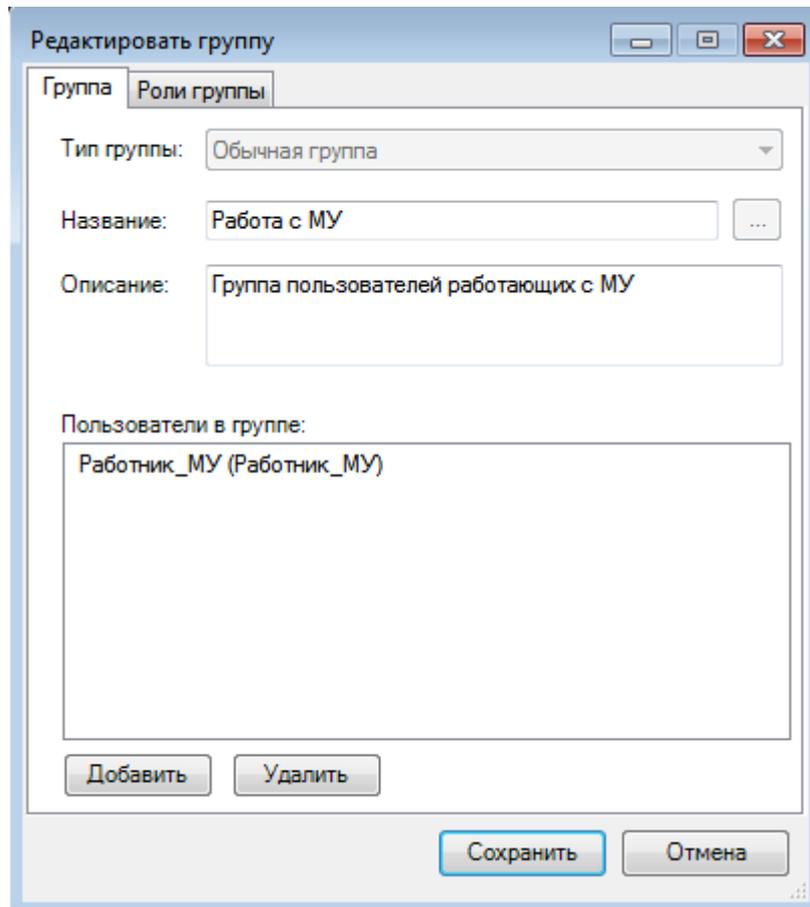


Рисунок 13. Диалог «Редактировать группу», вкладка «Группа»

2.1.3.3 Удаление группы пользователей

Для удаления группы пользователей нужно выбрать ее в списке и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 14) группа пользователей будет удалена. Выделив несколько записей, можно удалить несколько групп пользователей.

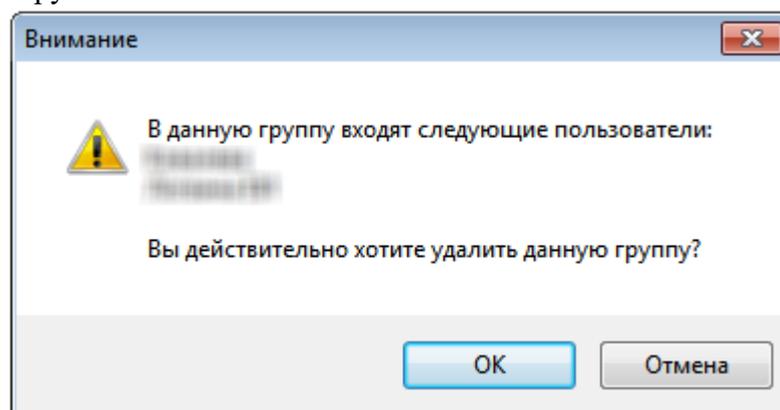


Рисунок 14. Диалог подтверждения удаления группы

2.1.4 Роли

Роль – именованный набор правил, определяющих доступ к элементам программы. При помощи грамотно настроенной роли можно разграничить доступ пользователя (или нескольких пользователей) к документам, отчетам, протоколам, библиотекам и отдельным функциям программы.

На вкладке «Роли» отображается список всех определенных в программе ролей (Рисунок 15). Список ролей пользователя также можно просмотреть в форме редактирования учетных записей (Рисунок 15) и на зависимой вкладке «Роли пользователей» при просмотре списка пользователей программы.

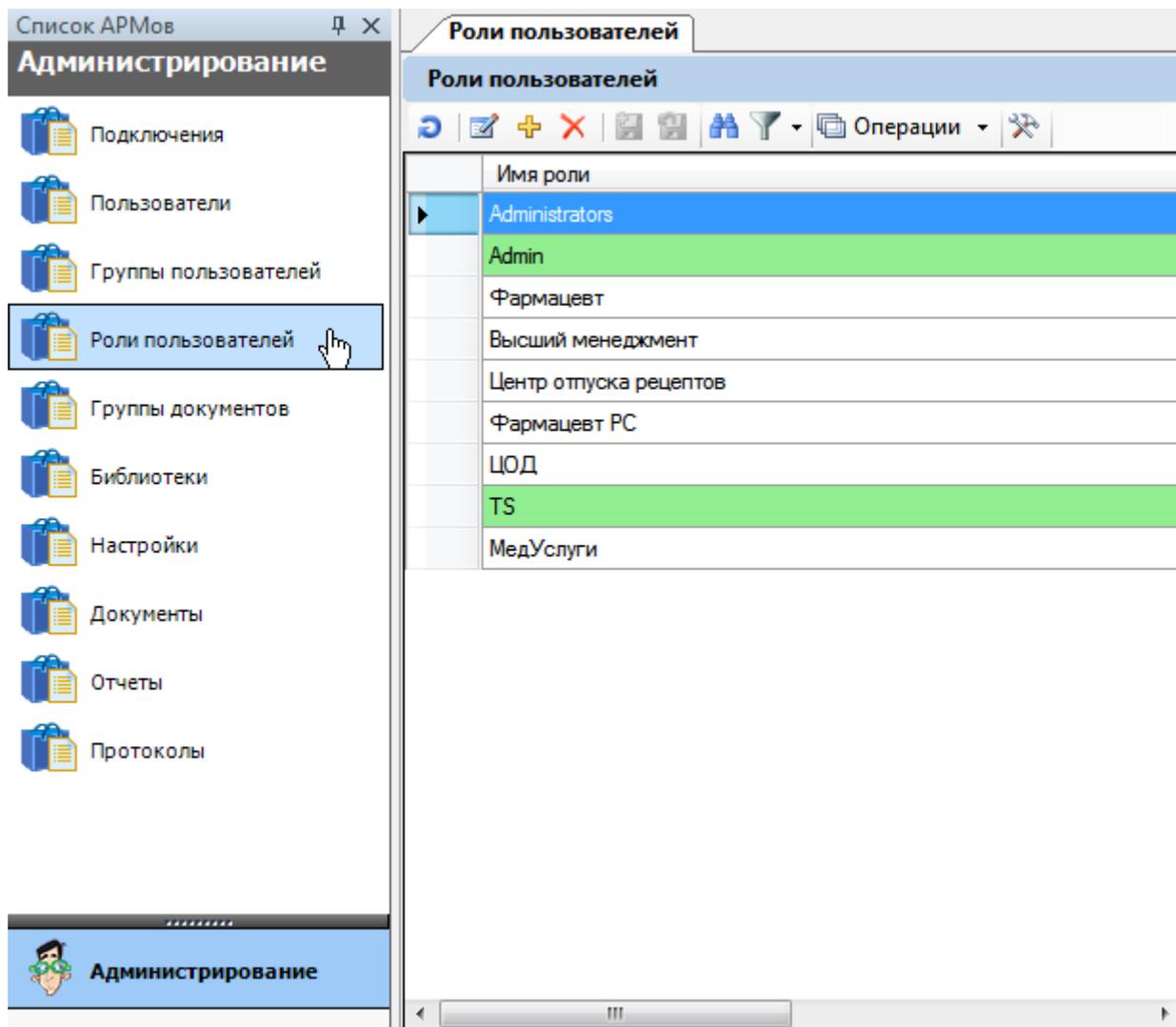


Рисунок 15. Список зарегистрированных ролей

2.1.4.1 Добавление роли

Добавление роли выполняется с помощью соответствующего диалога (Рисунок 16). При добавлении роли необходимо заполнить два обязательных поля:

- Имя роли – имя, под которым роль фигурирует в программе.
- Тип роли – атрибут, определяющий, на какие элементы программы распространяется данная роль. Может принимать значения «Пользовательская» и «Администраторская».

«Пользовательская» роль разграничивает права доступа в операционном приложении. «Администраторская» роль содержит права доступа для приложения администрирования. Пользователь, которому не назначена ни одна администраторская роль, не может выполнить вход в приложение администрирования. Подробнее смотрите п 2.1.11.

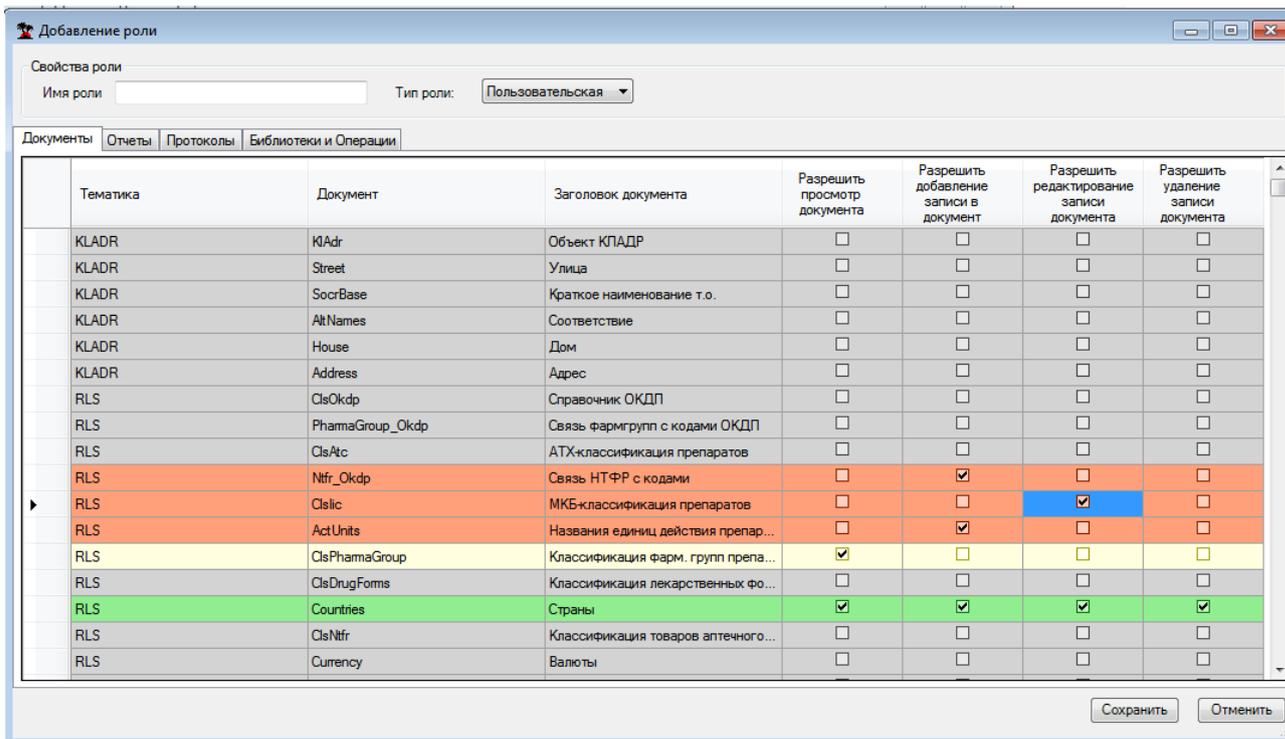


Рисунок 16. Диалог «Добавление роли», вкладка «Документы»

В зависимости от типа роли диалог содержит разный набор вкладок. Каждая из вкладок используется для настройки прав доступа к объектам определенного типа (документам, отчетам, протоколам и пр.). Для каждого типа объектов существует свой уникальный набор флагов. Доступ к конкретному объекту определяется набором флагов, выставленных в соответствующей строке таблицы.

К примеру, в пользовательской роли для конкретного типа документов можно разрешить просмотр, добавление записей, редактирование записей и удаление. Для отчетов и протоколов можно разрешить только их выполнение.

Настройка прав доступа для библиотек и операций находится на одной вкладке (Рисунок 17). В строках таблицы находятся записи подключенных библиотек (смотрите п 2.1.6), а в строках ниже библиотеки – операции, содержащиеся в ней. К примеру, на приведенном снимке экрана библиотека «OMS» содержит две операции – «Выгрузить» для документа «OMS.Error» и «Компилировать алгоритм» для документа «OMS.ExpertAlgorithm».

Отключение загрузки библиотеки приводит к отключению всех операций, содержащихся в ней. В частности, на приведенном снимке экрана отключена библиотека «SprAutoUpdate» и ее операция «Обновление справочных данных». Недоступные для изменения записи операций выделены курсивом. В случае, когда курсивом выделена запись библиотеки, это сигнализирует об ошибке загрузки подключенной библиотеки. Настройка прав доступа для такой библиотеки невозможна.

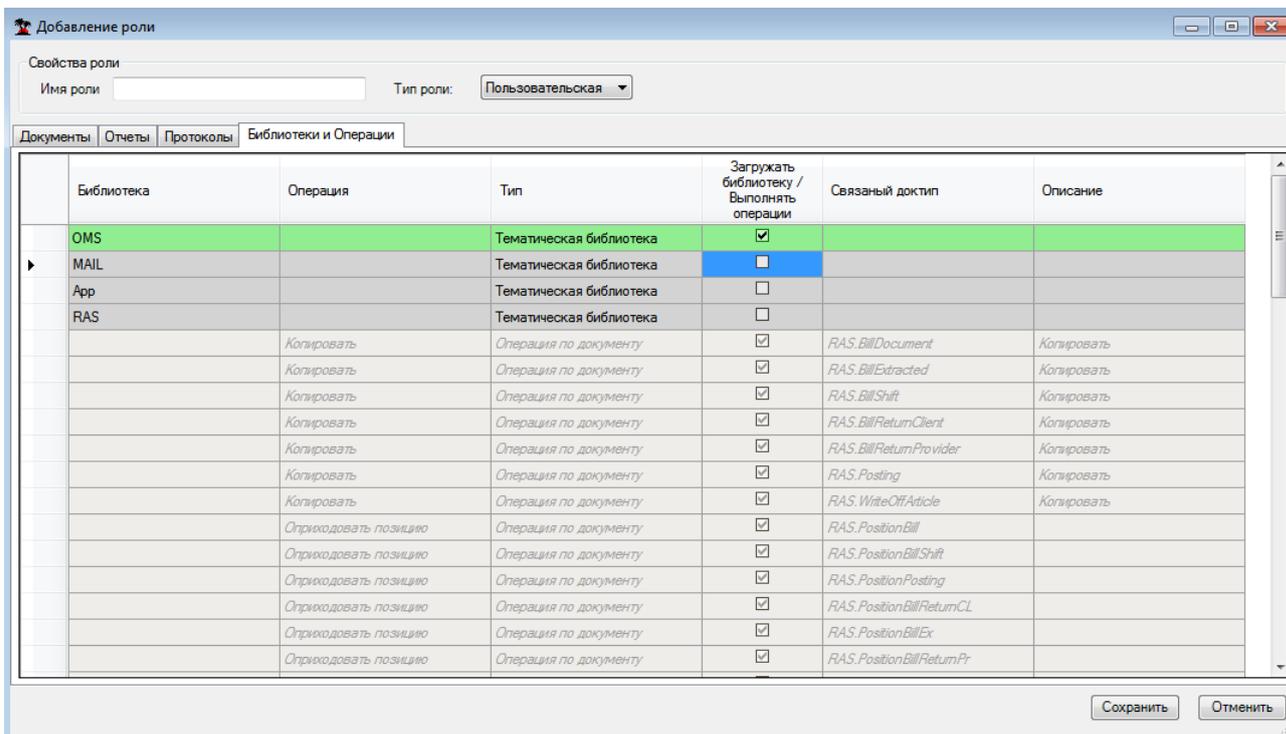


Рисунок 17. Диалог «Добавление роли», вкладка «Библиотеки и Операции»

У всех типов объектов существует флаг «Использовать данное правило», определяющий, задано ли правило доступа для конкретного объекта. Если данный флаг не установлен, для данной сущности права доступа не определены. В этом случае программа принимает решение о доступе к объекту с помощью правил системы прав доступа (описание представлено в п. 2.1.11).

2.1.4.2 Редактирование роли

Редактирование ролей выполняется в том же диалоговом окне, что и добавление (Рисунок 16). В случае если в результате редактирования роли будет изменен, ее тип, все записи о правах доступа другого режима будут удалены. При сохранении пользователь получит соответствующее предупреждение (Рисунок 18).

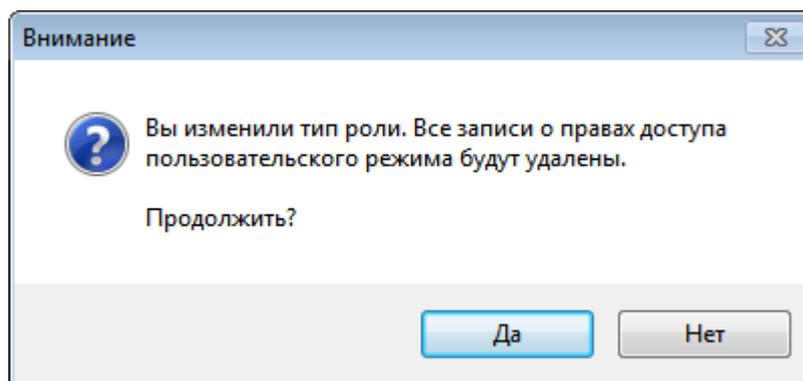


Рисунок 18. Предупреждение при изменении типа роли

2.1.4.3 Удаление роли

Для удаления роли нужно выбрать ее в списке и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 19) роль будет удалена. Выделив несколько записей, можно удалить несколько ролей.

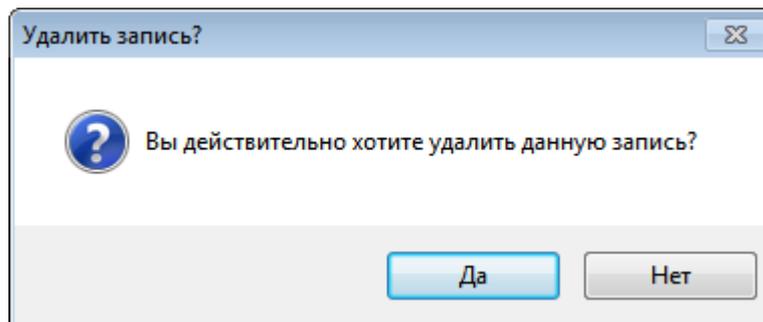


Рисунок 19. Диалог подтверждения удаления роли

2.1.5 Группы документов

Группы документов используются для группировки документов в операционном приложении. На вкладке «Группы» отображается список всех определенных в программе групп (Рисунок 20). В данном окне можно создавать новые группы, редактировать уже существующие и удалять ненужные.

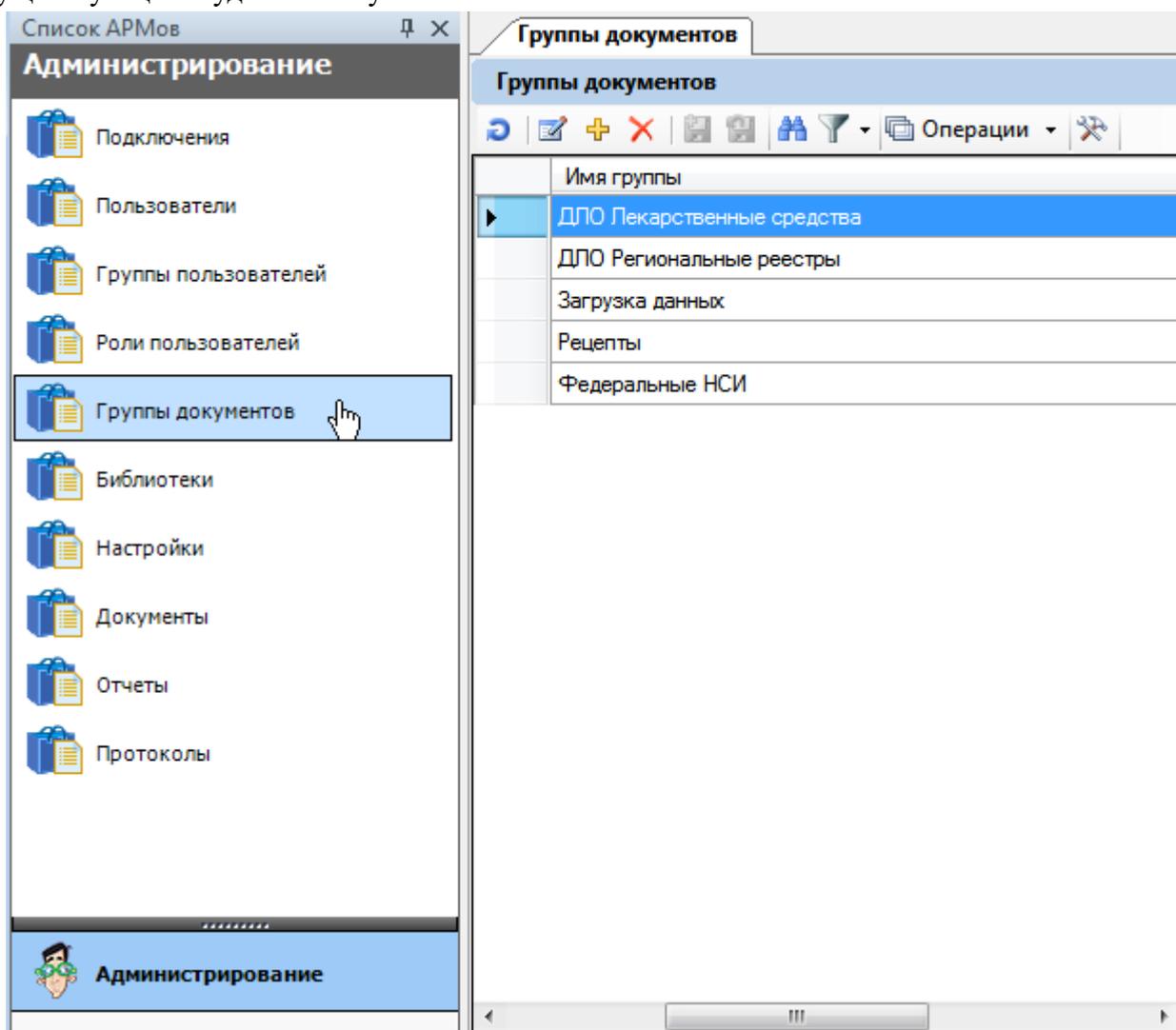


Рисунок 20. Вкладка «Группы»

2.1.5.1 Добавление группы

Добавление групп выполняется с помощью соответствующего диалога (Рисунок 21). Для группы необходимо указать имя, место расположения в операционном приложении и набор документов, входящих в нее.

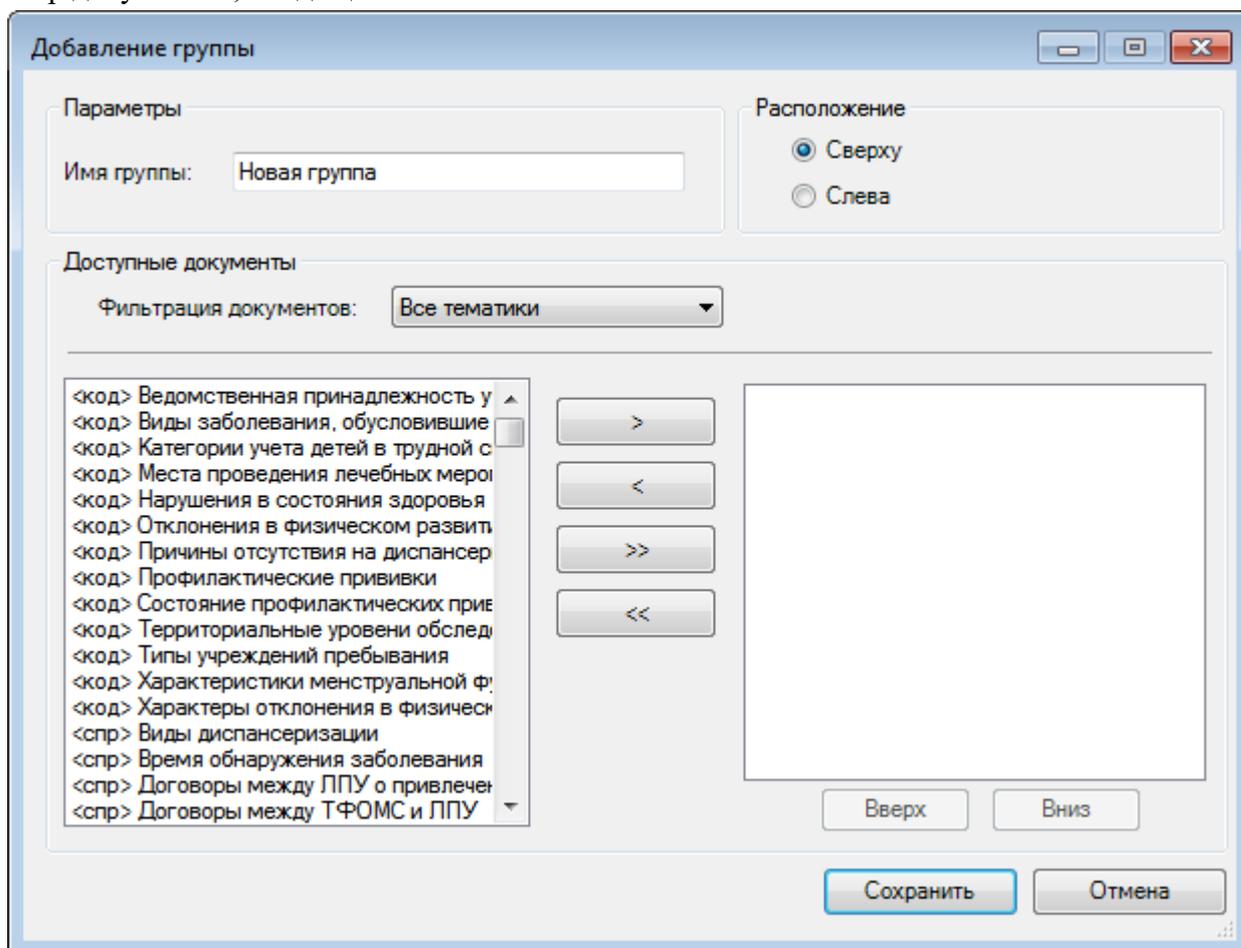


Рисунок 21. Диалог создания группы

Выбор документов для группы выполняется перемещением их из списка левой части окна в список правой. Список доступных документов можно фильтровать по тематике. В списке документов группы с помощью кнопок «Вверх» и «Вниз» можно определить порядок документов.

2.1.5.2 Редактирование группы

Редактирование группы выполняется в том же диалоговом окне, что и добавление (Рисунок 21). Все параметры доступны для изменения.

2.1.5.3 Удаление группы

Для удаления группы нужно выбрать ее в списке и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 22) группа будет удалена. Выделив несколько записей, можно удалить несколько групп.

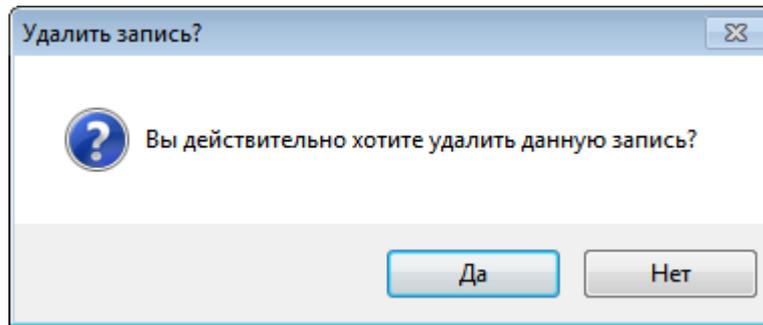


Рисунок 22. Подтверждение удаления группы

2.1.6 Библиотеки

На вкладке «Библиотеки» отображается список всех подключенных библиотек (Рисунок 23). В ней можно добавлять (подключать) новые библиотеки, редактировать уже существующие и удалять ненужные.

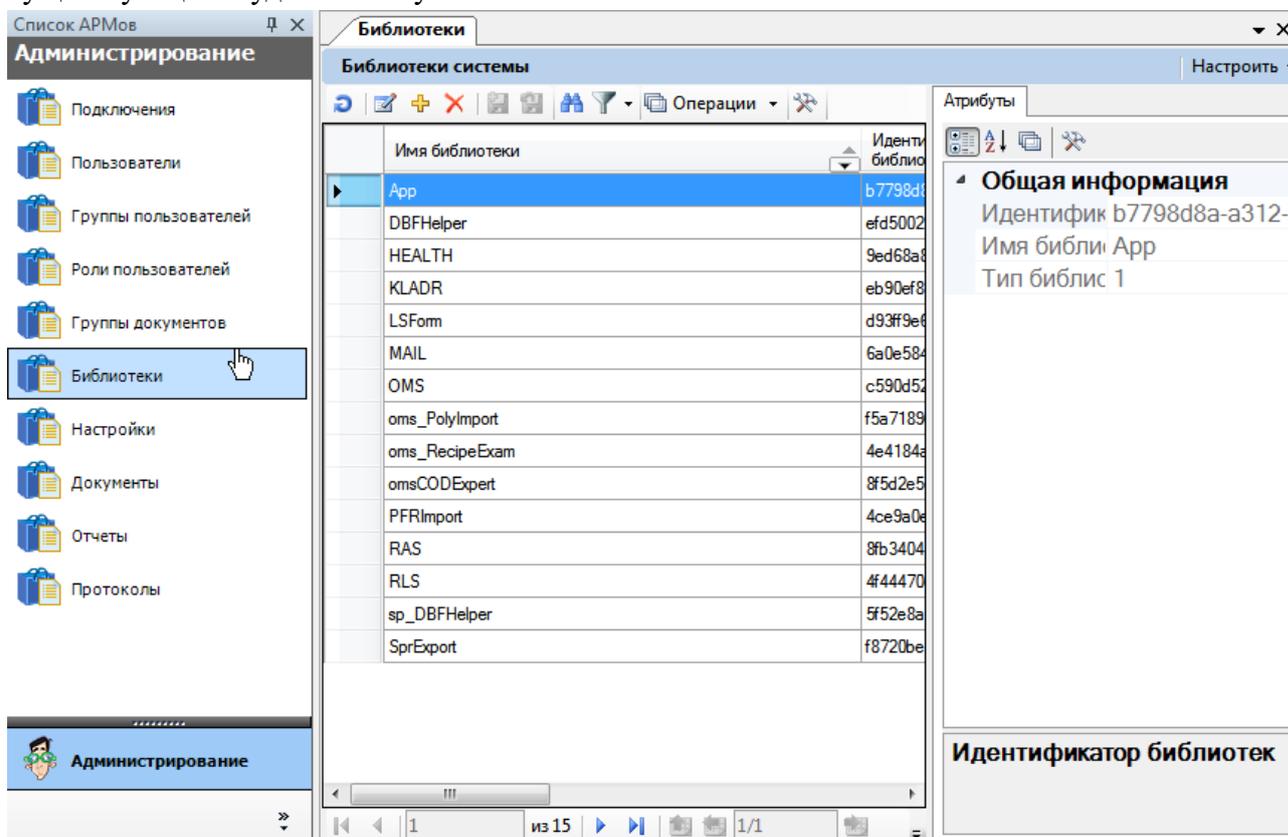


Рисунок 23. Вкладка «Библиотеки»

2.1.6.1 Подключение библиотек

Подключение (добавление) библиотеки производится в диалоговом окне (Рисунок 24).

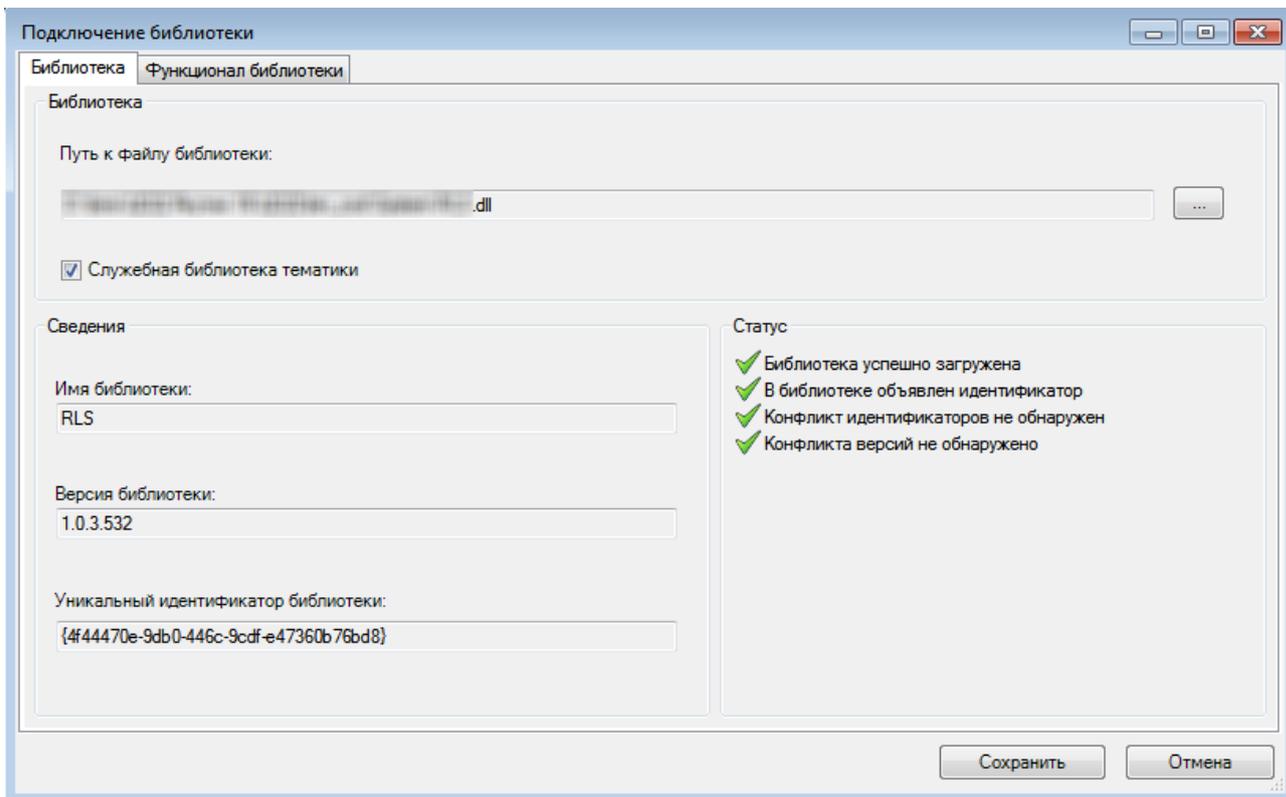


Рисунок 24. Диалог подключения библиотеки

Для подключения библиотеки необходимо выбрать подключаемый файл (как правило, из каталога System) и сохранить в базе данных сведения о нем. Флаг «Тематическая библиотека» необходимо установить для служебных тематических библиотек.

Прочие элементы диалога показывают состояние библиотеки, ее свойства, например, наличие обязательного идентификатора, удалось ли загрузить файл библиотеку в память, и пр.

2.1.6.2 Редактирование записи библиотеки

Редактирование записи библиотеки ничем не отличается от подключения и выполняется в том же диалоговом окне (Рисунок 24).

2.1.6.3 Отключение библиотек

Отключение библиотеки приводит к тому, что операционное приложение не использует функции, объявленные в этой библиотеке. Непосредственно сам файл библиотеки не удаляется. Для отключения библиотеки нужно выбрать ее в списке и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 25) библиотека будет отключена. Таким же образом можно выбрать и отключить несколько библиотек за один раз.

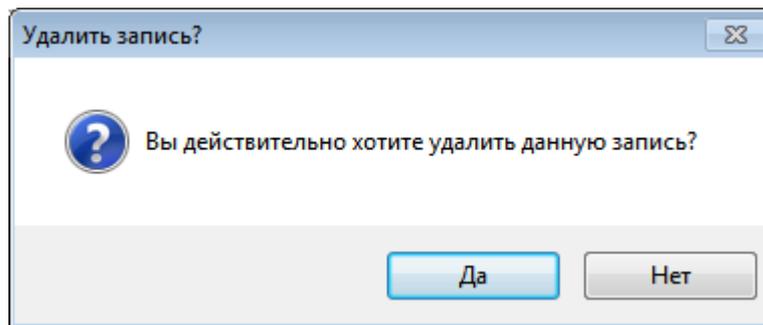


Рисунок 25. Подтверждение отключения библиотеки

2.1.7 Настройки

Для хранения разнородных данных используются пользовательские настройки, хранящиеся в базе данных. Каждая настройка идентифицируется именем, типом, уникальным идентификатором и принадлежностью к документу. Настройки могут хранить строковые, текстовые, двоичные, целочисленные значения, а также дату.

В разделе «Настройки» (Рисунок 26) отображается список всех определенных в программе пользовательских настроек.

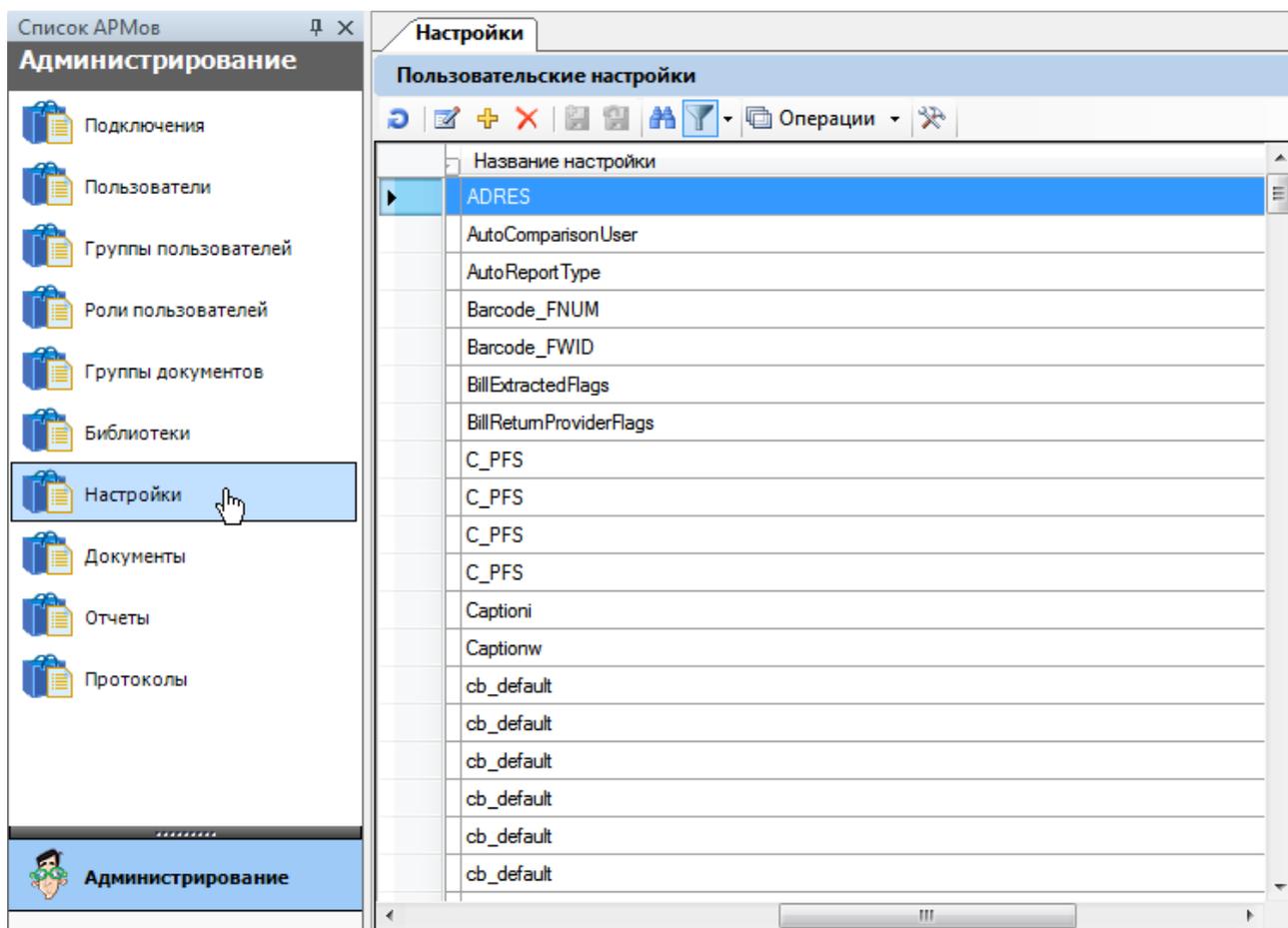


Рисунок 26. Вкладка пользовательских настроек

Настройки можно добавлять, редактировать и удалять, однако управление настройками требуется не часто, и требует от администратора четкого понимания выполняемых им действий.

2.1.7.1 Добавление пользовательских настроек

Добавление настройки выполняется в диалоговом окне (Рисунок 27).

Рисунок 27. Диалог «Добавление пользовательской настройки»

Поля ввода, значения которых необходимо задать:

- Название – имя создаваемой настройки.
- Тип настройки – значение поля определяет, какого типа данные могут быть в этой настройке сохранены (Рисунок 28).

Рисунок 28. Список типов настройки

- Документ – тип документа, к которому будет относиться данная настройка. Выбирается из списка, где перечислены все зарегистрированные типы документов в программе.
- Идентификатор владельца – уникальный идентификатор владельца настройки. Поле необязательно для заполнения.
- Пользователь – пользователь, от имени которого создается настройка. Пользователя можно выбрать только из списка уже определенных в программе пользователей. Если настройку создает встроенный в систему пользователь «Admin», такая настройка автоматически будет действовать и на всех остальных пользователей программы.

Если же настройку создал любой другой пользователь, то данная настройка будет действовать только для данного пользователя.

В зависимости от типа настройки, необходимо задать ее значение на одной из вкладок.

Строка – строковый тип. В поле вводится строка – непосредственно значение настройки (Рисунок 29).

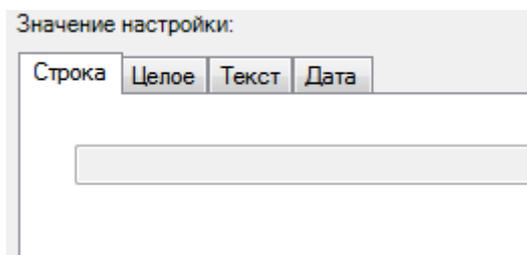


Рисунок 29. Пользовательская настройка – Вкладка «Строка»

Целое – представляет собой целое положительное значение, либо отрицательное число, или ноль. Значение можно либо непосредственно ввести в поле с клавиатуры, либо ввести, используя стрелки, расположенные справа от поля, «прокручиванием» значения до необходимой величины (Рисунок 30).

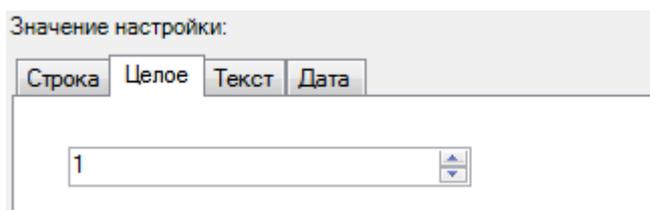


Рисунок 30. Пользовательская настройка – Вкладка «Целое»

Текст – значение представляет собой текстовые данные. Его можно ввести непосредственно в поле ввода, либо загрузить из файла. Значение также можно сохранить в файл. Загрузка и сохранение текста осуществляются с помощью стандартных диалогов открытия и сохранения файлов (Рисунок 31).

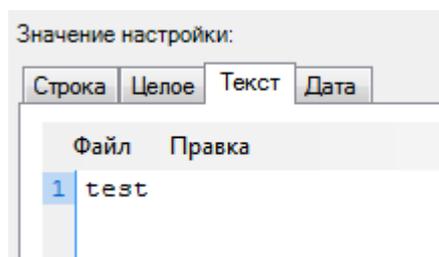


Рисунок 31. Пользовательская настройка – Вкладка «Текст»

Его можно ввести непосредственно в поле ввода, либо загрузить из файла (Рисунок 32).

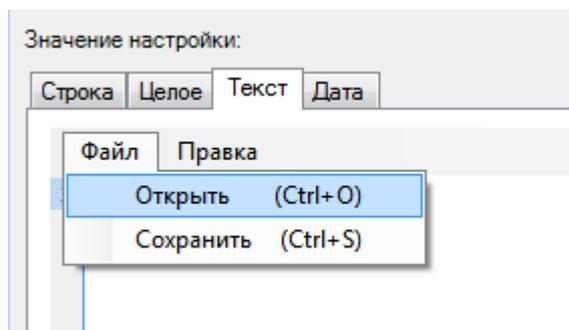


Рисунок 32. Открытие файла для загрузки текста

Значение также можно сохранить в файл. Загрузка и сохранение текста осуществляются с помощью стандартных диалогов открытия и сохранения файлов (Рисунок 33).

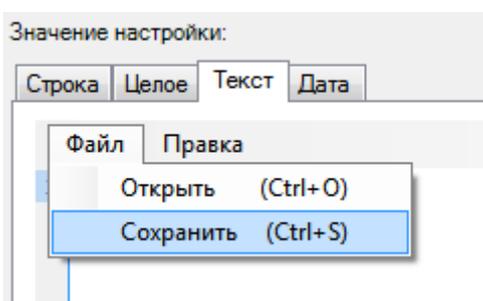


Рисунок 33. Сохранение текста в файл

Дата – значение представляет собой дату. По умолчанию отображается текущая дата. Чтобы изменить дату, используется стандартный диалог выбора даты (Рисунок 34).

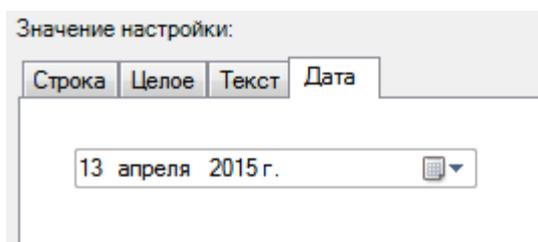


Рисунок 34. Пользовательская настройка – Вкладка «Дата»

2.1.7.2 Редактирование пользовательских настроек

Редактирование пользовательских настроек выполняется в том же диалоговом окне, что и добавление (Рисунок 27). При редактировании настройки невозможно изменить ее тип. Все прочие параметры доступны для изменения.

2.1.7.3 Удаление пользовательских настроек

Для удаления настройки необходимо выбрать ее в списке и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 35) настройка будет удалена. Выделив несколько записей, можно удалить несколько настроек.

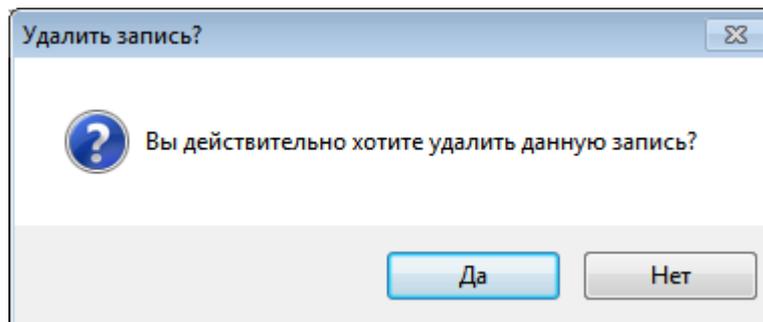


Рисунок 35. Подтверждение удаления настройки

2.1.8 Документы

Вкладка «Документы» содержит набор зарегистрированных в программе типов документов. Приложение администрирования не позволяет управлять составом и структурой документов. Однако на вкладке «Документы» в контекстном меню находится управление индексами (Рисунок 36). «Индекс» – сущность базы данных, позволяющая ускорить операции поиска. Приложение администрирования позволяет управлять индексами, а также выполнять переиндексацию базы данных.

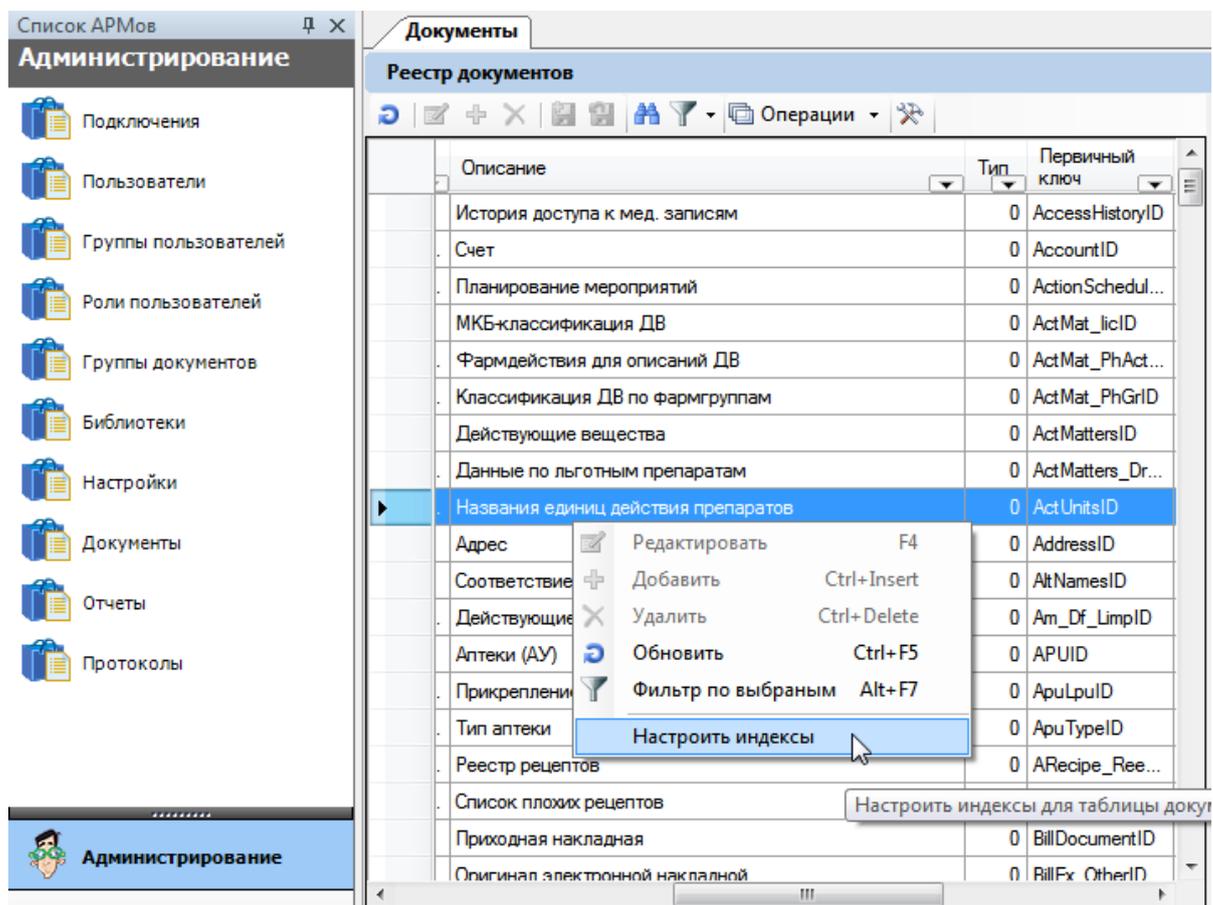


Рисунок 36. Вкладка «Документы»

2.1.8.1 Настройка индексов

Настройка индексов производится для конкретного документа с помощью Менеджера индексов (Рисунок 37). Вызвать «Менеджер индексов» можно из контекстного меню, подпункт «Настроить индексы» (Рисунок 36).

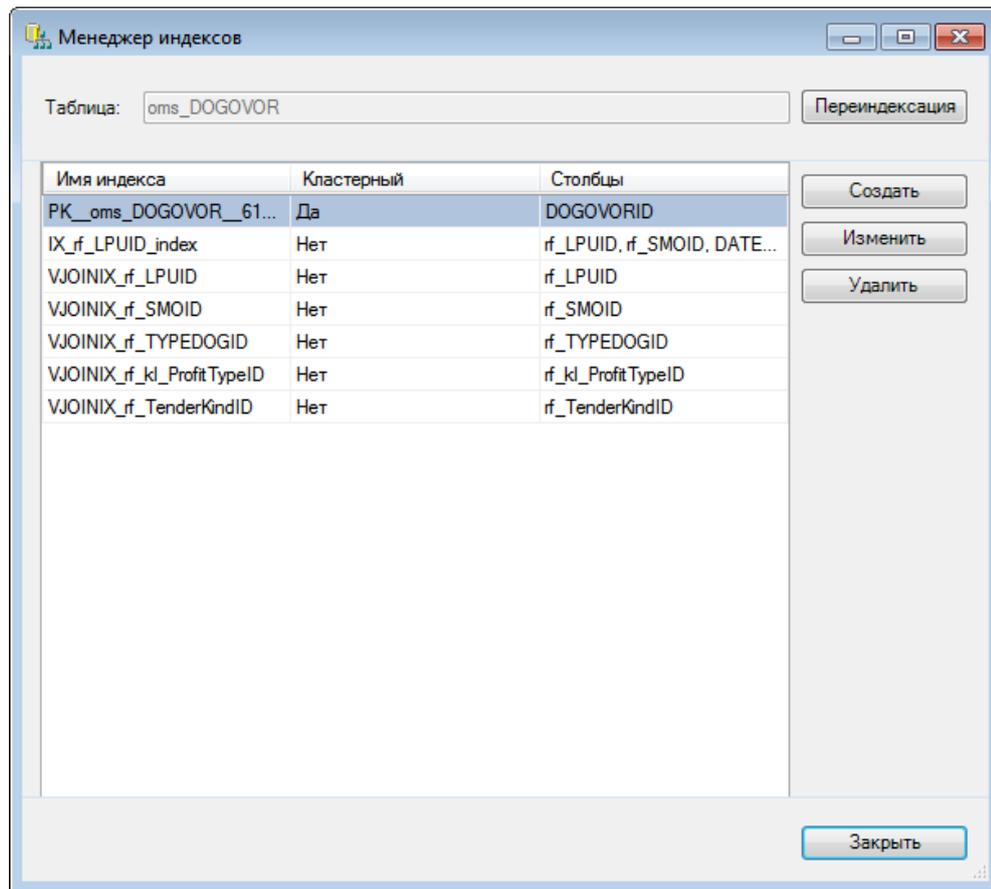


Рисунок 37. Менеджер индексов

Менеджер индексов позволяет создавать, удалять, редактировать индексы, проводить переиндексацию таблицы. Создание и редактирование индексов выполняется в одном и том же диалоговом окне (Рисунок 38).

На вкладке «Колонки» (Рисунок 38) необходимо:

- ввести имя индекса;
- выбрать индексируемые столбцы;
- установить порядок индексируемых столбцов (с помощью кнопок «Вверх», «Вниз»);
- выбрать направление сортировки для каждого выбранного столбца.

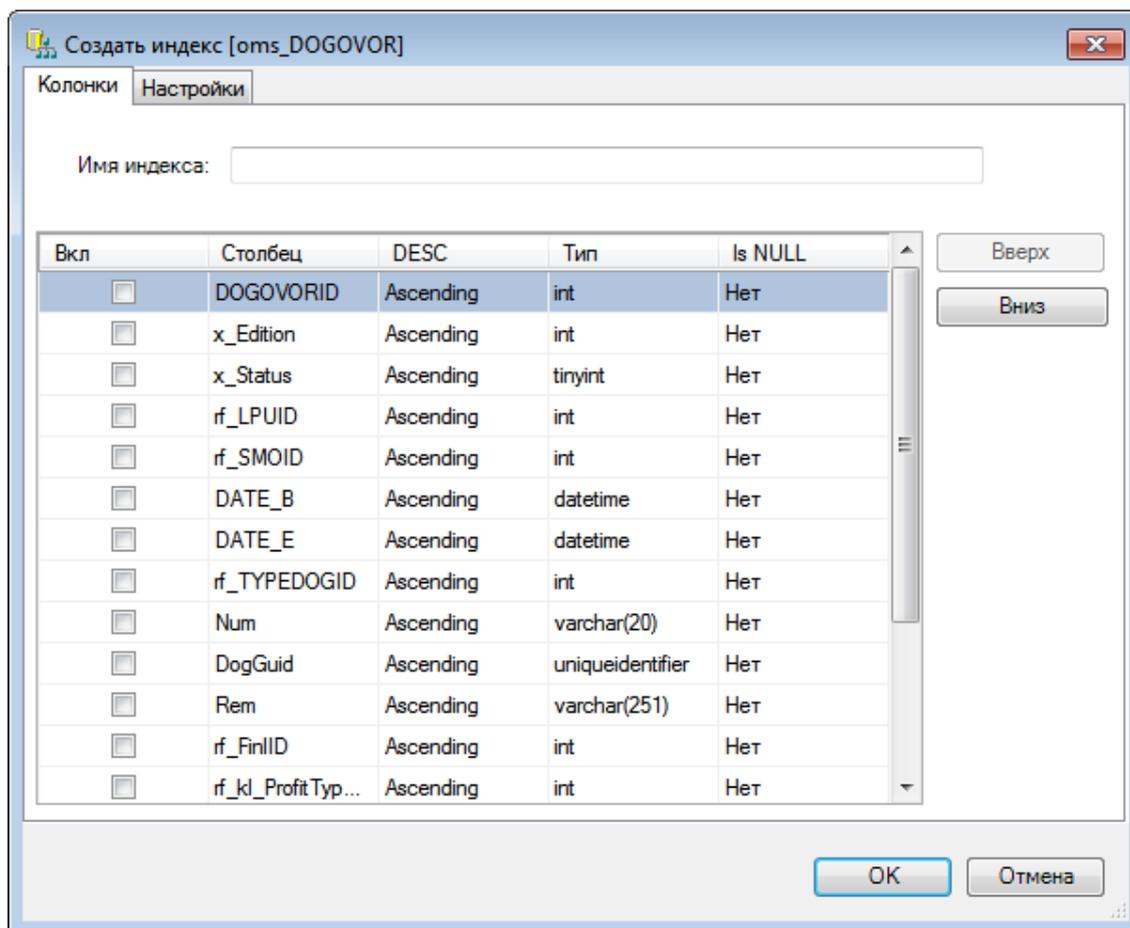


Рисунок 38. Вкладка «Колонки» в диалоге создания индексов

На вкладке «Настройки» (Рисунок 39) необходимо установить свойства индекса:

- Кластерный индекс (может быть только 1 для каждой таблицы).
- Уникальные значения – значения индексируемых столбцов должны быть уникальны.
- Не пересчитывать статистику – пересчитывать или не пересчитывать статистику автоматически.
- Уровень заполнения – установить уровень заполнения индекса.
- Удалять существующий – автоматически удалять существующий с таким же именем индекс при создании нового.

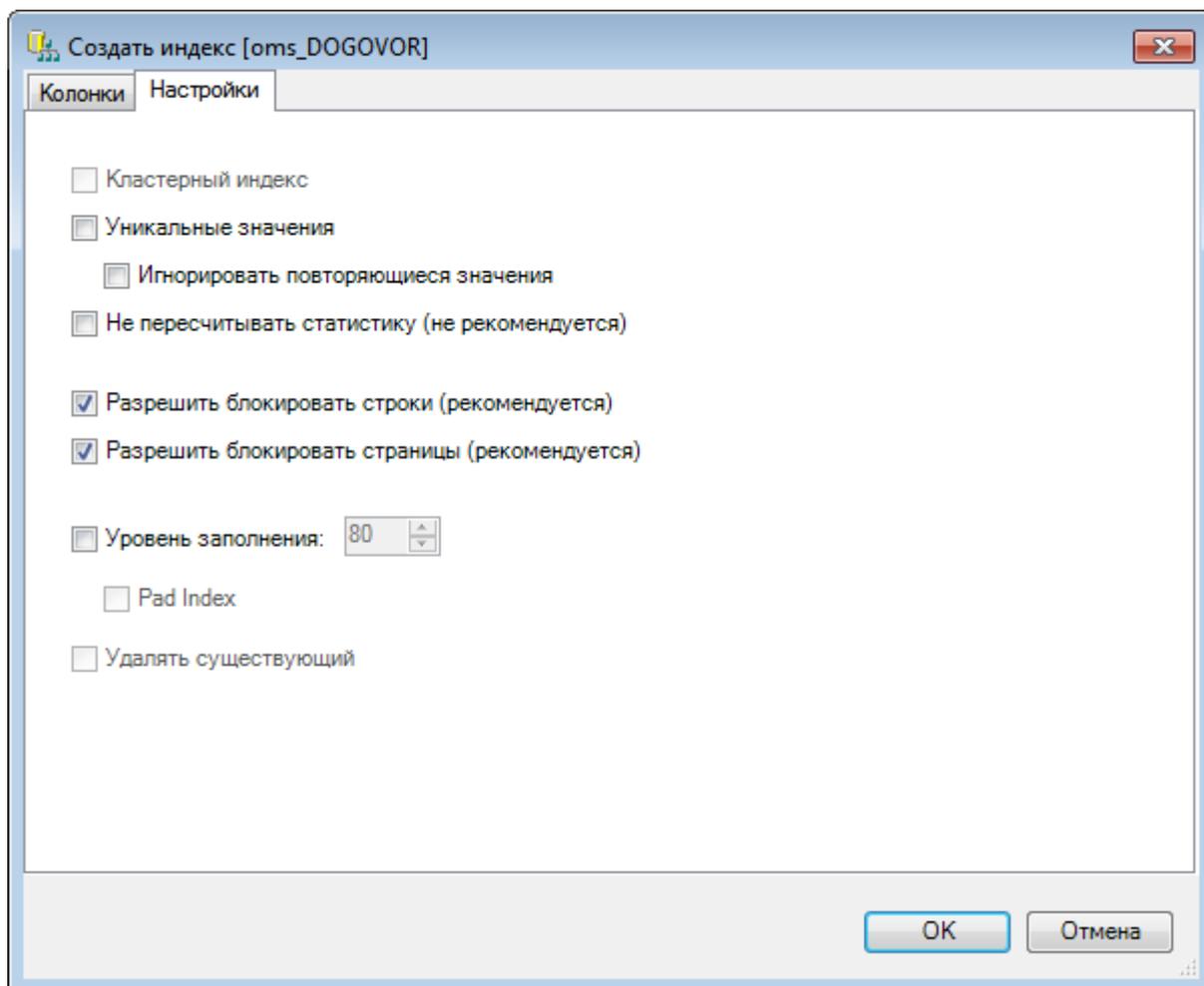


Рисунок 39. Вкладка «Настройки» в диалоге создания индексов

При нажатии кнопки «Переиндексация» происходит переиндексация таблицы и появляется информационное сообщение (Рисунок 40).

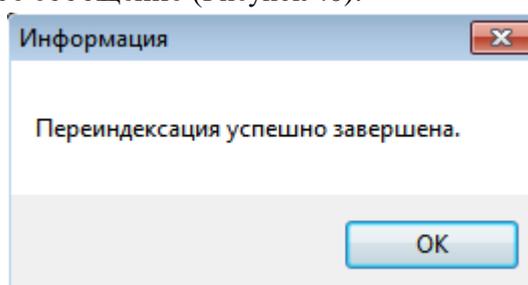


Рисунок 40. Информационное сообщение

Удаление индексов также выполняется в менеджере при нажатии кнопки «Удалить». Подробнее о свойствах индексов и управлении ими обратитесь к документации по СУБД MSSQL.

2.1.9 Отчеты

Одной из важнейших составляющих платформы является модуль работы с отчётами. Он позволяет создавать отчётные документы практически любой сложности. Платформа позволяет работать с двумя разными типами отчётов: стандартными и определёнными.

«Стандартный отчет» применим к текущему экземпляру документа произвольного типа, и не требует предварительного описания. Форма стандартного отчета получается

автоматически на основе единого для всех типов документов шаблона и описания типа документа.

Более сложные «определенные отчеты» предусматривают процедуру подготовки схемы формирования отчета. Схемы определенных отчетов представляют собой экземпляры особой сущности, которые хранятся в качестве документов типа «Отчет» в БД. Также эти отчеты могут быть выгружены (загружены) в файлы специального формата. Управление определенными отчетами выполняется в приложении администрирования на вкладке «Отчеты» (Рисунок 41).

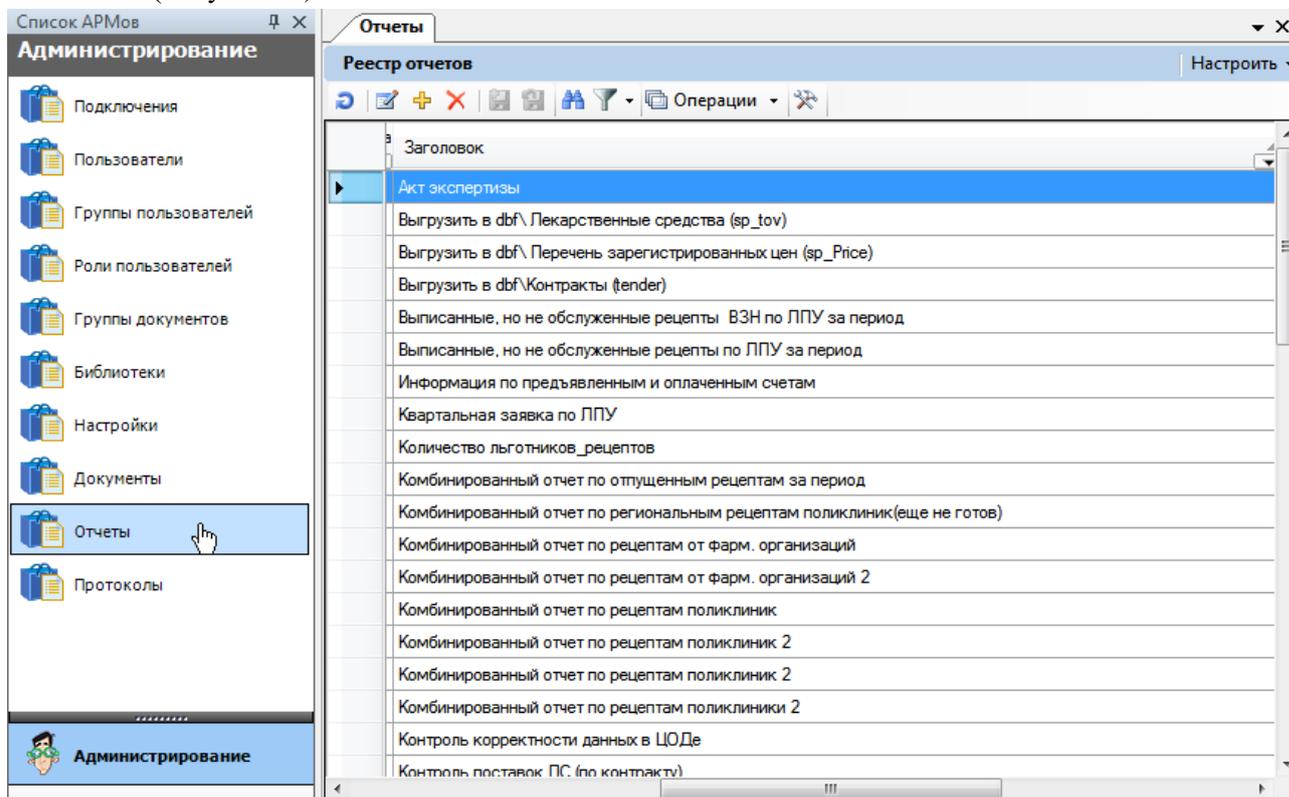


Рисунок 41. Вкладка «Отчеты»

При нажатии на кнопку «Операции» в меню управления записями откроется окно для выбора операции над отчетами (Рисунок 42). Отчеты могут быть выгружены (загружены) в файлы специального формата.

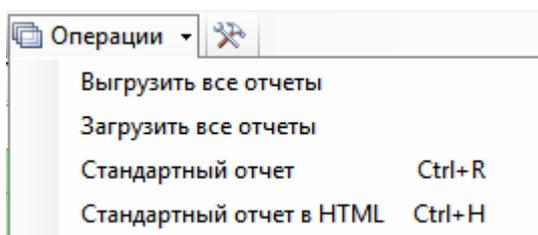


Рисунок 42. Окно выбора «Операции»

2.1.9.1 Добавление отчетов

Добавление отчетов можно выполнить с помощью диалога, либо загрузить отчет из файла (Рисунок 43). Операция загрузки расположена в контекстном меню списка отчетов. Диалог позволяет создать отчет на основе собственного шаблона. Отчет может быть привязан к конкретному документу, либо нет; иметь различный формат, и пр. Создание отчетов требует владения приложением Microsoft Excel, знания языка T-SQL, а также

базовых навыков работы с Visual Basic for Applications. Подробно о создании отчетов смотрите Приложение А настоящего документа.

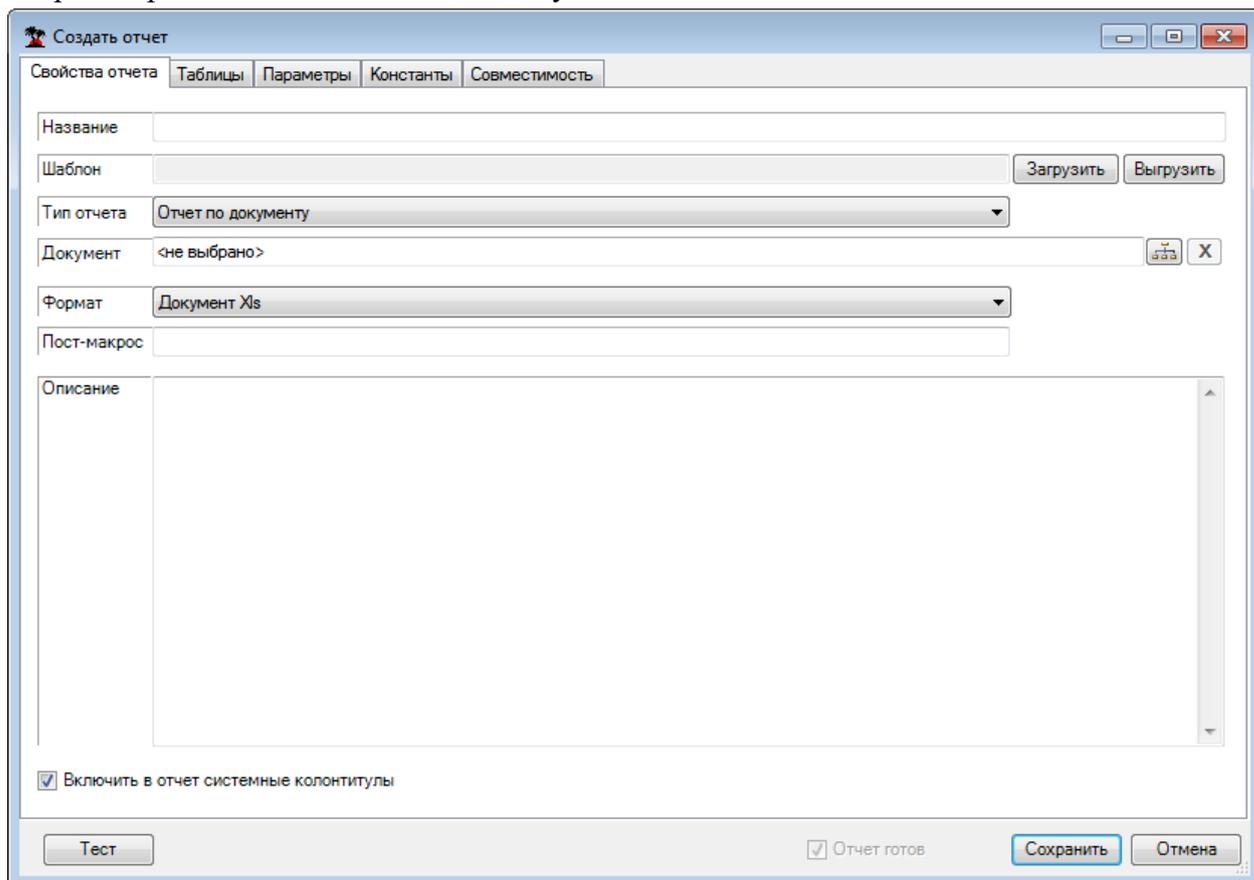


Рисунок 43. Добавление (редактирование) отчетов

2.1.9.2 Редактирование отчетов

Редактирование отчетов выполняется в том же диалоговом окне, что и добавление (Рисунок 43). Все параметры доступны для изменения. Подробно о редактировании отчетов смотрите Приложение А настоящего документа.

2.1.9.3 Удаление отчетов

Для удаления отчета нужно выбрать его в списке и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 44) группа будет удалена. Выделив несколько записей, можно удалить несколько отчетов.

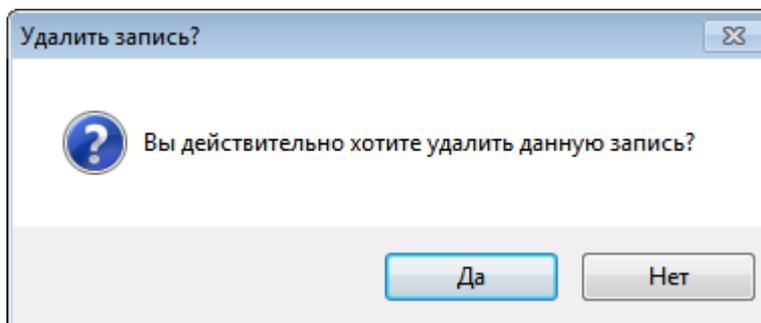


Рисунок 44. Подтверждение удаления отчета

2.1.10 Протоколы

Протокол – сценарий загрузки или выгрузки данных, хранящийся в базе данных. Операции над протоколами производятся через приложение администрирования. Чтобы начать работу с протоколами, зайдите на вкладку «Протоколы» (Рисунок 45).

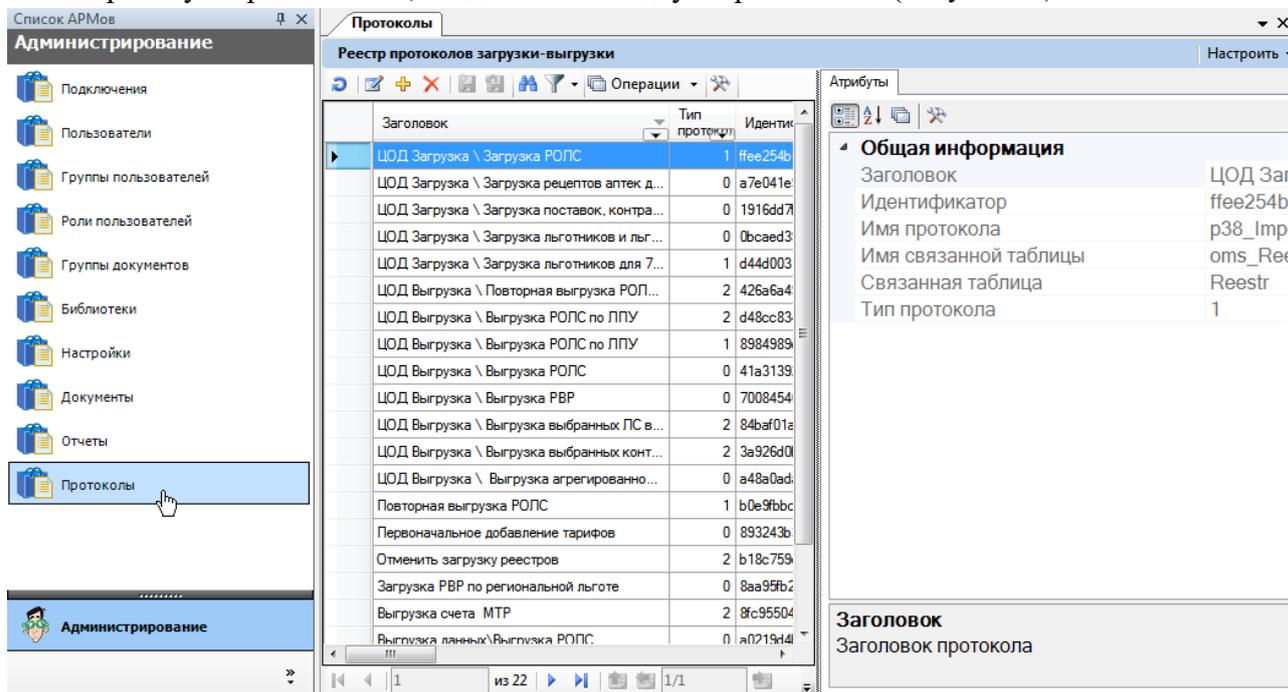


Рисунок 45. Вкладка «Протоколы»

2.1.10.1 Добавление протоколов

Для создания нового или загрузки существующего протокола выберите «Добавить» в контекстном меню, либо воспользуйтесь соответствующей кнопкой. Будет открыт диалог редактирования протокола, в котором можно создать протокол или загрузить его из файла (Рисунок 46). Подробно о создании протоколов смотрите Приложение Б настоящего документа.

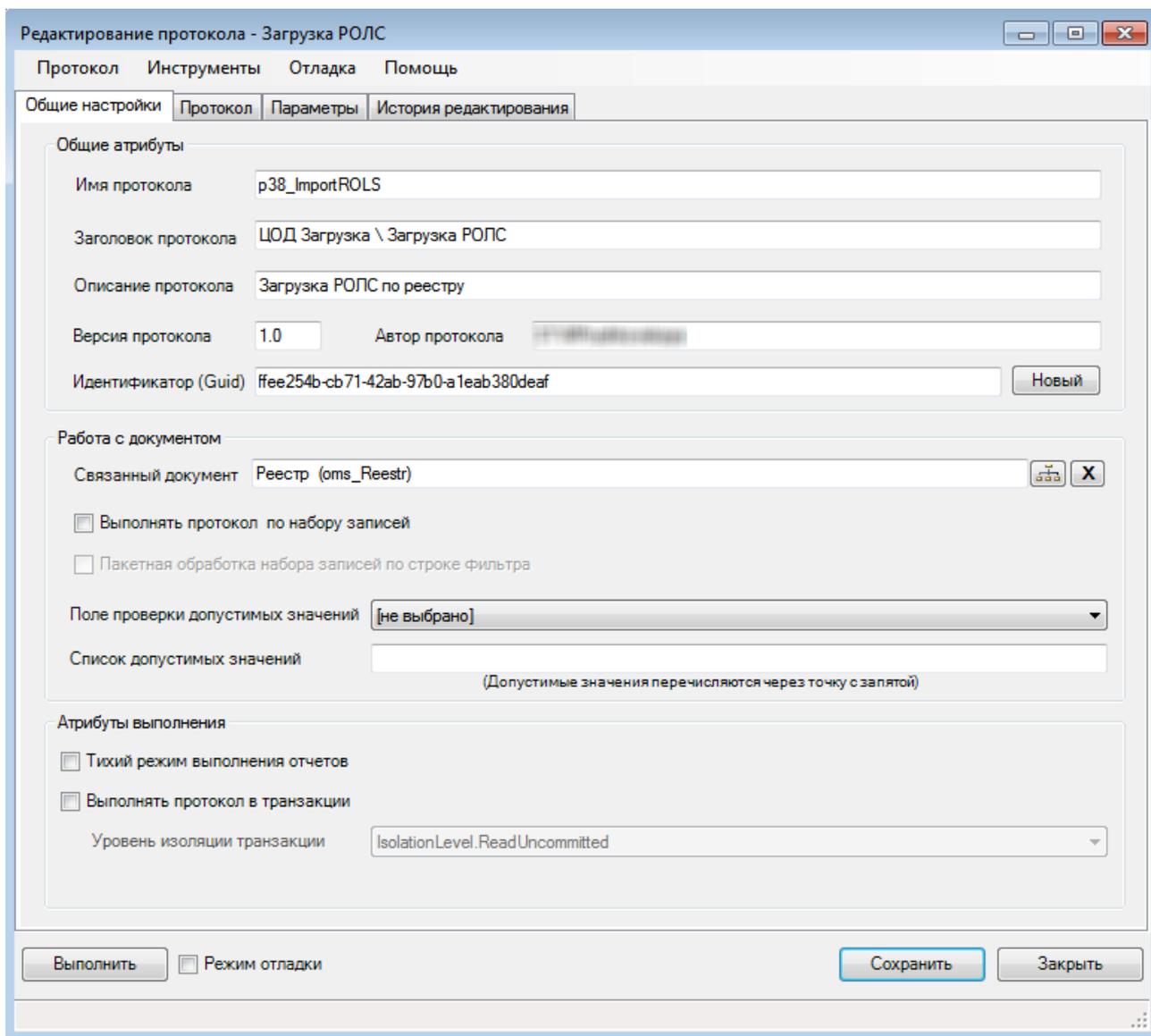


Рисунок 46. Добавление (редактирование) протокола

2.1.10.2 Редактирование протоколов

Редактирование протокола выполняется в том же диалоговом окне, что и добавление (Рисунок 46). Подробно о создании протоколов смотрите Приложение Б настоящего документа.

2.1.10.3 Удаление протоколов

Для удаления протокола нужно выбрать его в списке и нажать соответствующую кнопку, либо воспользоваться контекстным меню. После подтверждения (Рисунок 47) группа будет удалена. Таким же образом можно выбрать и удалить несколько протоколов за один раз.

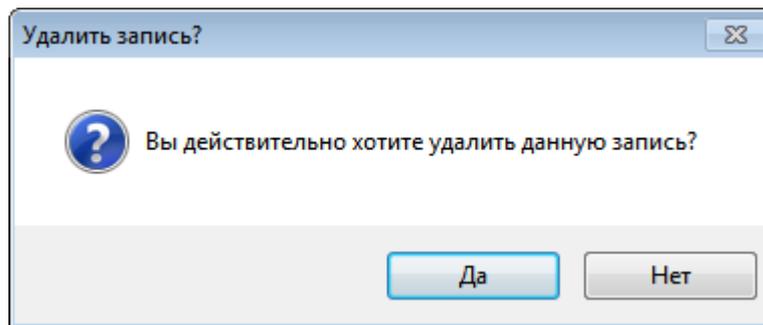


Рисунок 47. Подтверждение удаления протокола

2.1.11 Права доступа

Система разграничения прав доступа – это группа правил, используемых программой для определения набора операций, доступных конкретному пользователю. Для системы прав доступа введены понятия «суперпользователь», «роль» и «тип роли».

Для входа в программу с правами суперпользователя необходимо воспользоваться служебной учетной записью «Admin». Удаление данной записи запрещено. Настройка ролей в данной учетной записи недоступна. Пользователь «Admin» всегда обладает полными правами на все элементы программы. Редактирование (в том числе и переименование) служебной записи «Admin» разрешено.

Роль – именованный набор правил, определяющих доступ к элементам системы. Каждому пользователю может быть назначено произвольное число ролей (и даже не назначено ни одной).

Тип роли – атрибут, указывающий, на какие элементы системы распространяется данная роль. Может принимать значения: «Пользовательская» и «Администраторская». «Пользовательская» роль разграничивает права доступа в операционном приложении. «Администраторская» роль содержит права доступа для приложения администрирования. Пользователь, которому не назначена ни одна администраторская роль, не может выполнить вход в приложение администрирования.

На основе системы ролей в контексте конкретного пользователя элемент системы может иметь одно правило доступа, набор из нескольких правил доступа, или не иметь ни одного правила вовсе. В соответствии от вида ситуации применяются:

1. Права по умолчанию.
2. Заданные права.
3. Результирующие права.

2.1.11.1 Права по умолчанию

Отсутствие информации о правах доступа на объект возможно в двух ситуациях:

1. У пользователя отсутствуют роли.
2. Во всех ролях пользователя отсутствуют активные правила для данного объекта.

В подобных случаях к объекту системы применяются права по умолчанию, а именно правило – «Что не разрешено – запрещено».

2.1.11.2 Заданные права

Объект системы может иметь одно правило доступа в случае:

1. Пользователю назначена одна роль с активным правилом для данного объекта.

2. Пользователю назначено несколько ролей, но для данного объекта существует активное правило только в одной из ролей.

В ситуации с существованием одного правила, объект системы получает права доступа, указанные в данном правиле.

2.1.11.3 Результирующие права

В общем случае, когда пользователю назначено несколько ролей, объекту системы соответствует несколько назначенных прав. В этом случае объект получает права, подчиняющиеся следующей логике: Если во всех правилах некая операция над объектом разрешена – она будет разрешена. В противном случае эта операция запрещена.

2.2 Подпункты меню «Операции»

В подпункте меню «Операции» находятся различные служебные функции, необходимые для настройки и диагностики приложения. Часть инструментов, представленных в данном меню, может воспользоваться только специалист, четко представляющий архитектуру системы.

2.2.1 Диагностика

Для проведения диагностических работ программы необходимо выбрать один из пунктов меню «Операции» → «Диагностика» (Рисунок 48).

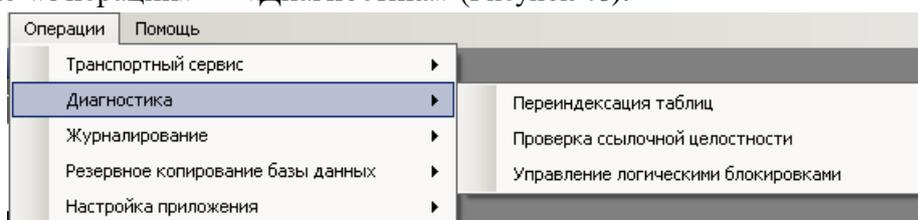


Рисунок 48. Окно выбора операции

2.2.1.1 Переиндексация таблиц

Переиндексация – процесс перестроения индексов таблиц БД. Переиндексация нужна для уменьшения фрагментации индекса, поэтому ее необходимо регулярно проводить в целях повышения быстродействия БД.

Для проведения переиндексации таблиц существует специальный диалог (Рисунок 49). Вызвать его можно из «Операции» → «Диагностика» → «Переиндексация таблиц».

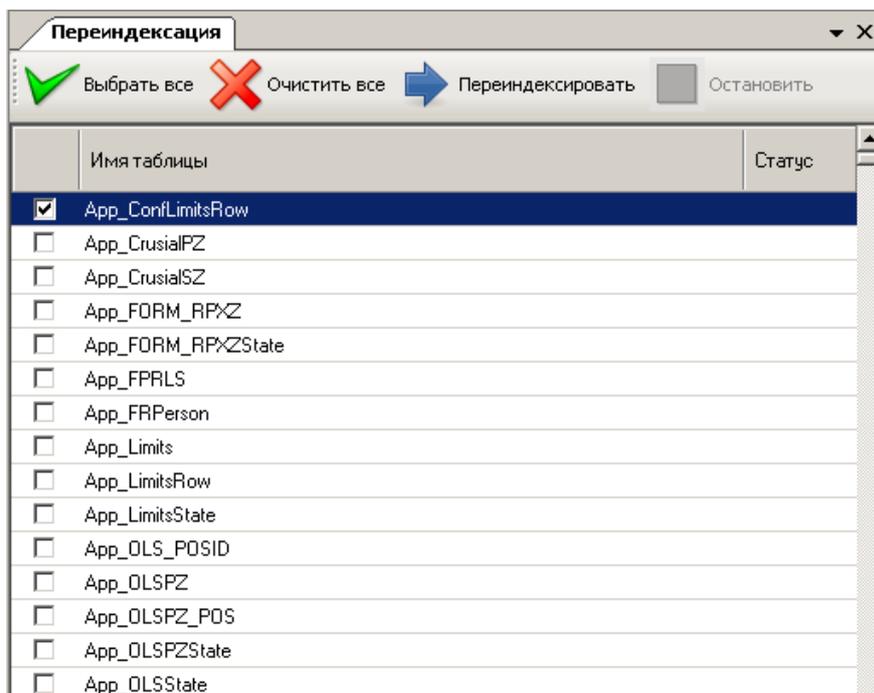


Рисунок 49. Диалог «Переиндексация»

В окне переиндексации следует выбрать из списка таблицы, нуждающиеся в переиндексации. Для начала операции следует нажать кнопку «Переиндексировать». Остановить процесс переиндексации можно, нажав кнопку «Остановить» (Рисунок 50). После остановки переиндексацию можно продолжить, нажав кнопку «Переиндексировать». Список таблиц, назначенных на переиндексацию, можно изменять в любой момент. Если нужно переиндексировать все таблицы, нужно выделить все таблицы в списке кнопкой «Выбрать все» и нажать кнопку «Переиндексировать». Для снятия выделения со всех выбранных таблиц нужно нажать кнопку «Очистить все».

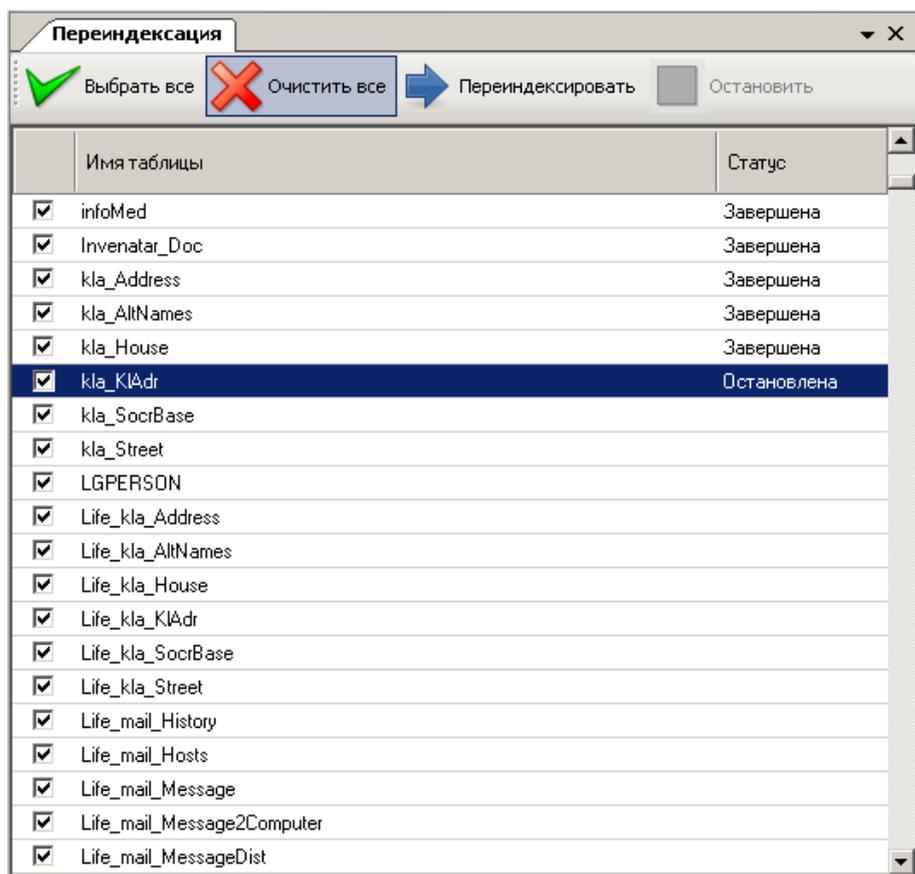


Рисунок 50. Диалог переиндексации – Остановка операции

2.2.1.2 Управление логическими блокировками

Блокировка – внутренний механизм системы. Функционирует без вмешательства пользователя. Однако, в случае сбоев аппаратуры, обрыва связи и прочих ситуаций возможно возникновение сбоев. В таком случае возможны ситуации, когда какой-либо документ останется заблокированным, что будет мешать нормальному функционированию системы.

Для просмотра и удаления подобных блокировок существует специальный диалог (Рисунок 51). Вызвать его можно из «Операции» → «Диагностика» → «Управление логическими блокировками».

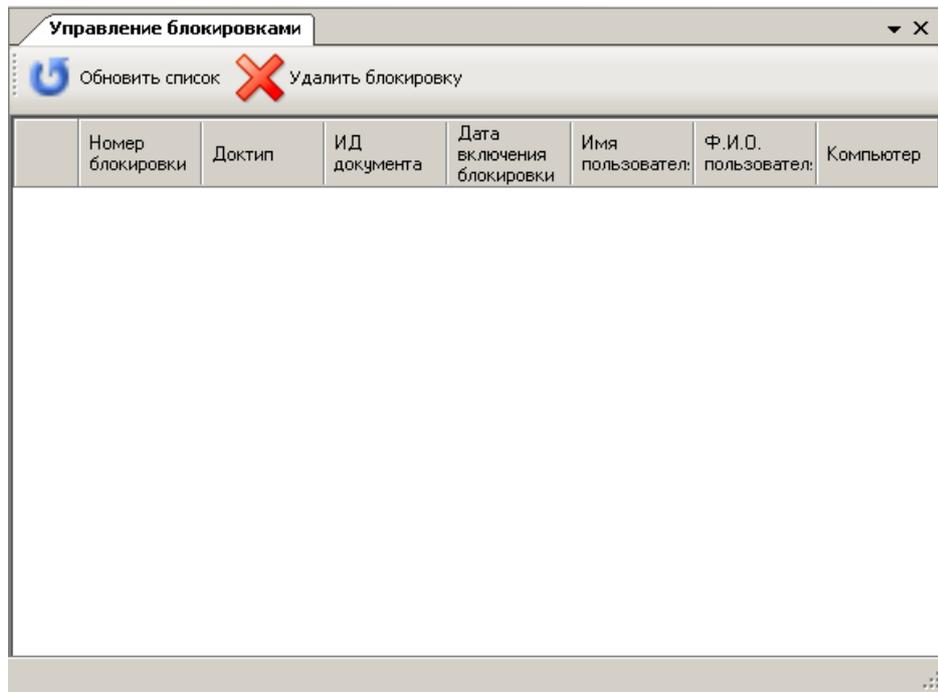


Рисунок 51. Диалог «Управление блокировками»

Диалог позволяет просмотреть оставшиеся блокировки и с помощью кнопки  удалить ненужные.

2.2.1.3 Проверка ссылочной целостности

Одним из элементов контроля целостности данных является диалог «Проверка ссылочной целостности». Вызвать его можно из «Операции» → «Диагностика» → «Проверка ссылочной целостности» (Рисунок 52).

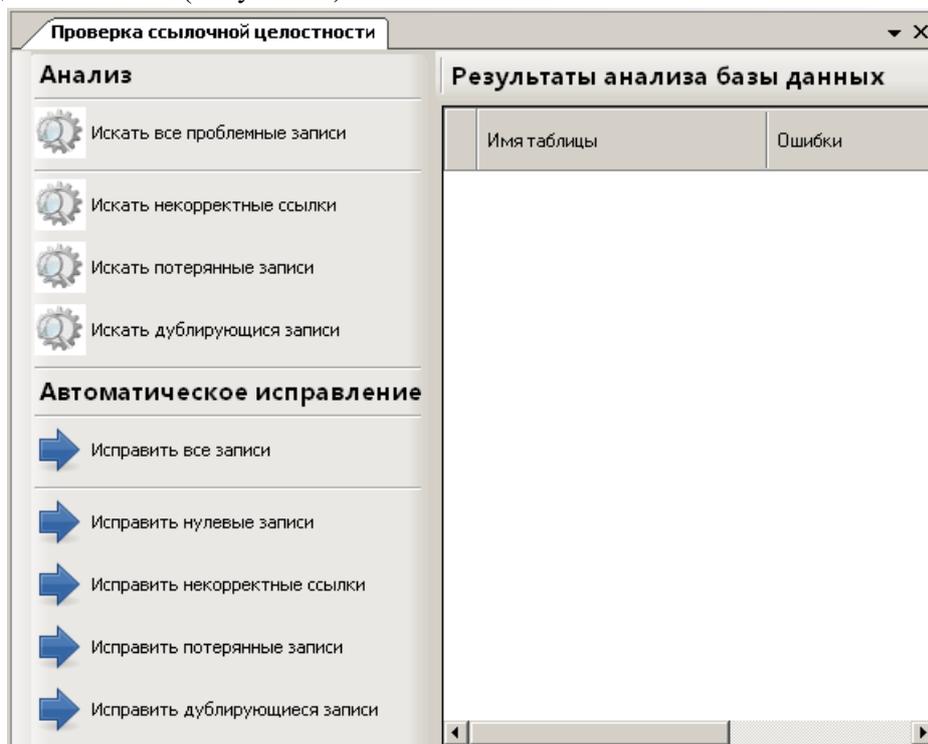


Рисунок 52. Проверка ссылочной целостности

Чтобы выполнить один из поисков проблемных записей, надо нажать одну из кнопок «Искать» Администратор проводит анализ БД, сообщает об окончании поиска (Рисунок 53) и предоставляет список некорректных записей (Рисунок 54).

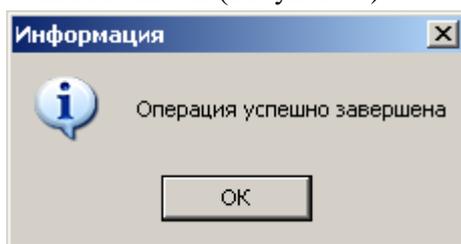


Рисунок 53. Окончание операции проверки ссылочной целостности

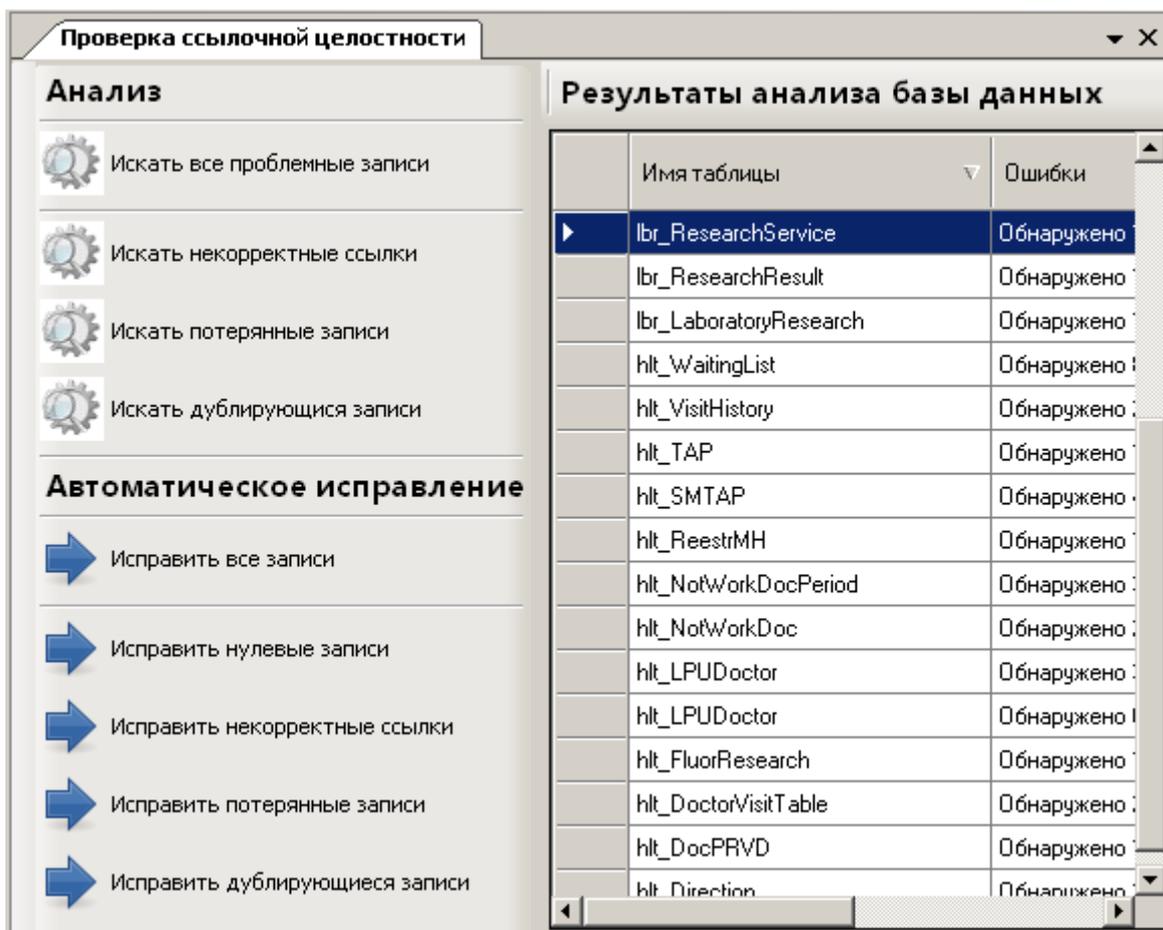


Рисунок 54. Результаты проверки ссылочной целостности

Чтобы выполнить автоматическое исправление ошибочных записей, надо нажать на одну из кнопок «Исправить» Администратор производит исправление ошибочных записей и сообщает об окончании операции. Если повторно выполнить поиск проблемных записей, то в окне «Результаты анализа базы данных» не должно быть записей с ошибками. Если же какие-то ошибочные записи остались, то надо запустить повторно режим автоматического исправления.

2.2.2 Журналирование

Доступ для просмотра и очистка журналов работы программы осуществляется в подпункте меню «Операции» → «Журналирование» (Рисунок 55).

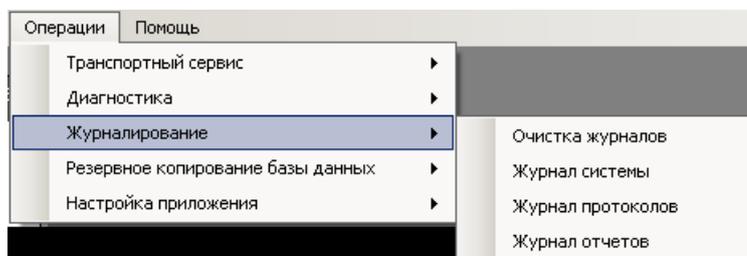


Рисунок 55. Окно выбора операции

2.2.2.1 Очистка журналов

Очистка журналов программы осуществляется в подпункте меню «Операции» → «Журналирование» → «Очистка журналов» (Рисунок 56).

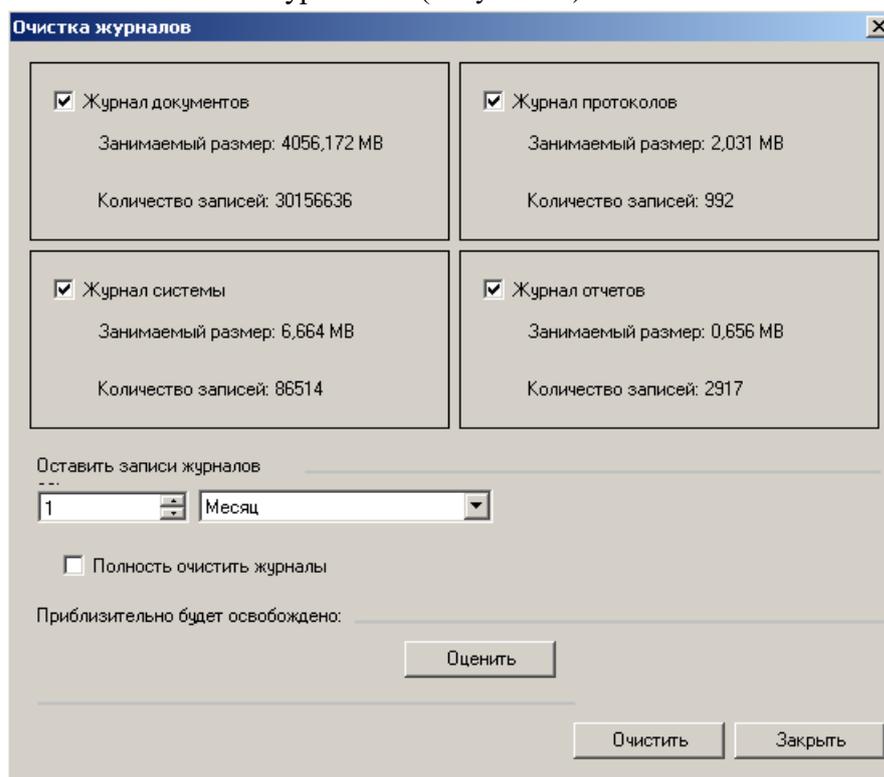


Рисунок 56. Окно очистки журналов

Для того чтобы предварительно оценить объем освобожденного пространства, можно перед очисткой журналов нажать на кнопку «Оценить» (Рисунок 57).

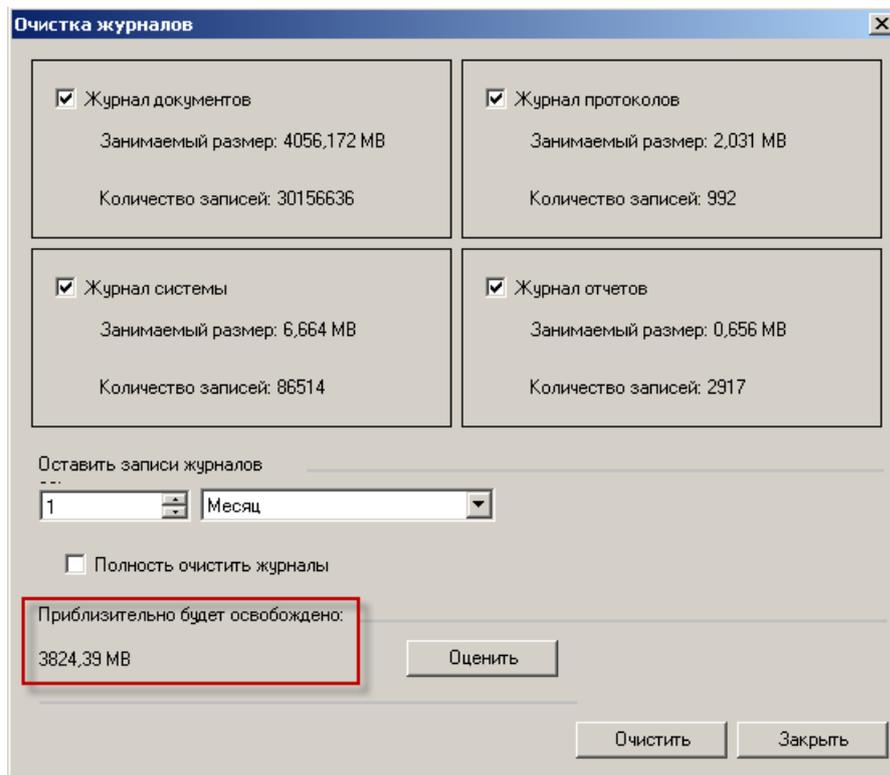


Рисунок 57. Окно очистки журналов

2.2.2.2 Журнал системы

Журнал системы – средство контроля над операциями, выполняемыми системой. Журнал доступен в подпункте меню «Операции» → «Журналирование» → «Журнал системы». Диалог позволяет просмотреть полную историю системных событий (Рисунок 58) и перечень выполненных операций конкретным пользователем (Рисунок 59).

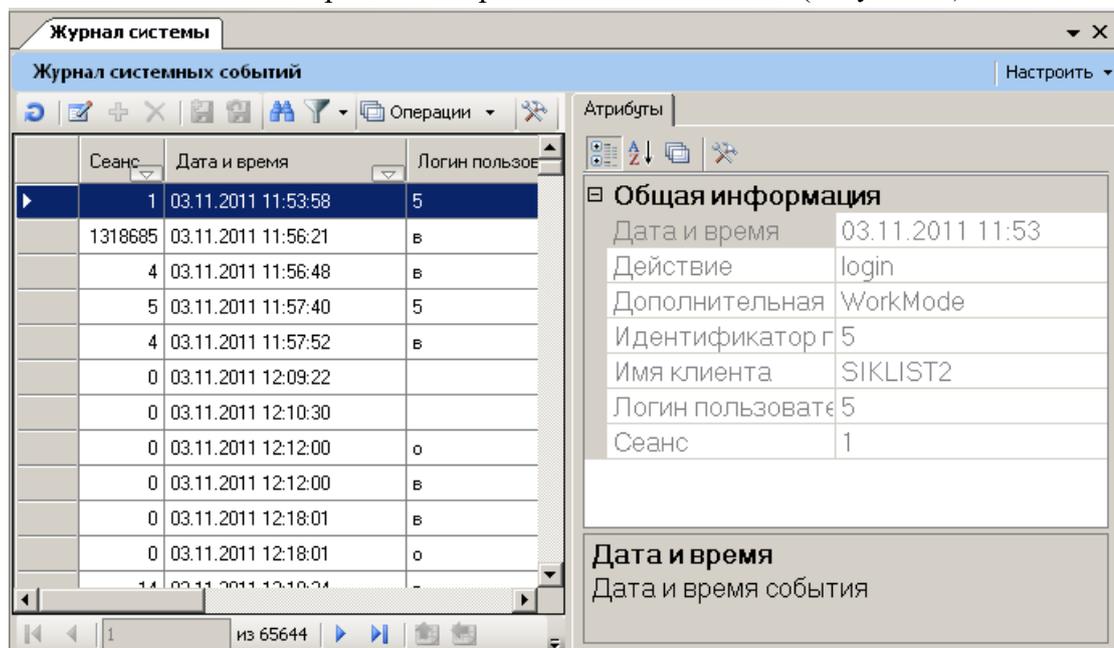


Рисунок 58. Журнал системных событий

Описание	Операция	Документ	Таблица
5 добавил запись в докум...	Вставка	Документ нетрудоспособ...	hlt_NotWorkDoc
5 добавил запись в докум...	Вставка	Период действия	hlt_NotWorkDocPeriod
5 отредактировал запись ...	Обновление	Период действия	hlt_NotWorkDocPeriod
5 отредактировал запись ...	Обновление	Документ нетрудоспособ...	hlt_NotWorkDoc
5 добавил запись в докум...	Вставка	Документ нетрудоспособ...	hlt_NotWorkDoc
5 добавил запись в докум...	Вставка	Период действия	hlt_NotWorkDocPeriod
5 отредактировал запись ...	Обновление	Период действия	hlt_NotWorkDocPeriod
5 отредактировал запись ...	Обновление	Документ нетрудоспособ...	hlt_NotWorkDoc
5 отредактировал запись ...	Обновление	Документ нетрудоспособ...	hlt_NotWorkDoc
5 добавил запись в докум...	Вставка	Документ нетрудоспособ...	hlt_NotWorkDoc
5 добавил запись в докум...	Вставка	Период действия	hlt_NotWorkDocPeriod
5 добавил запись в докум...	Вставка	Период действия	hlt_NotWorkDocPeriod
5 отредактировал запись ...	Обновление	Период действия	hlt_NotWorkDocPeriod
5 отредактировал запись ...	Обновление	Период действия	hlt_NotWorkDocPeriod

Рисунок 59. Журнал операций

Так же можно просмотреть информацию по конкретной проведенной операции данным пользователем (Рисунок 60).

Атрибут	Значение
Дата выдачи	03.11.2011 0:00:00
Закрыт	04.11.2011 0:00:00
Ф.И.О. врача, закрывшего листок	
По уходу	False
Возраст нуждающегося в уходе (год, мес.)	0,00
Пол нуждающегося в уходе	False
Флаг	0
Место работы	ООО Чол Косатка
Ф.И.О. нетрудоспособного	Сапрыкин Геннадий Иванович
Возраст нетрудоспособного	0
Пол нетрудоспособного	True
Адрес нетрудоспособного	
Приступить к работе	01.01.2222 0:00:00
Комментарий	
Направлен на МСЭ	01.01.1900 0:00:00
Код врача, закрывшего листок	
Ф.И.О. нетрудоспособного	

Рисунок 60. Просмотр сведений об измененной записи в журнале операций

2.2.2.3 Журнал обновлений

Журнал обновлений – средство просмотра обновлений, выполняемых системой. Журнал доступен в подпункте меню «Операции» → «Журналирование» → «Журнал обновлений». Диалог позволяет просмотреть историю обновлений (Рисунок 61).

Пакет обновления	Идентификатор пакета	Описание	Статус	Изменен
bf0241f2-8133-444f-a67e-20493943974	bf0241f2-8133-444f-a67e-20493943974e	Исправлены отчеты "Реестр выписанных рецептов по ЛПУ" и "Паспорт врачебного участка граждан (Форма №030-Р-П)_2304".	Загружен	27.11.2013 9:24:28
bf815f55-2366-40f5-bc52-f01e0df36d88	bf815f55-2366-40f5-bc52-f01e0df36d88	Исправлена ошибка с правами записи при составлении расписания. Выведен реестр врачей по ВЗН в группу регистры.	Загружен	27.11.2013 9:24:28
c1008172-5569-4d85-b7b6-beffc391cb12	c1008172-5569-4d85-b7b6-beffc391cb12	Разрешен ввод листов нетрудоспособности с пустой серией.	Загружен	27.11.2013 9:24:27
cb16b7b3-bc19-448a-bed2-4d75f6c222ca	cb16b7b3-bc19-448a-bed2-4d75f6c222ca	Исправлен отчет "Выписанные, но не отпущенные рецепты"	Загружен	27.11.2013 9:24:26
ce22ad40-4724-4bc1-8d09-4e172c937e57	ce22ad40-4724-4bc1-8d09-4e172c937e57	Обновление справочных данных (REFS 308).	Загружен	27.11.2013 9:24:26
ce380f81-ce8a-445d-9df5-2f1b5535bbd6	ce380f81-ce8a-445d-9df5-2f1b5535bbd6	Исправлена ошибка с правами записи при составлении расписания. Выведен реестр врачей по ВЗН в группу регистры.	Загружен	27.11.2013 9:24:28
ce58661a-d7de-4695-8918-ca37e298bc76	ce58661a-d7de-4695-8918-ca37e298bc76	Описание для файла обновления не задано	Загружен	27.11.2013 9:24:25
d1681418-4161-48c0-9b87-e73d527ace1e	d1681418-4161-48c0-9b87-e73d527ace1e	Описание для файла обновления не задано	Загружен	27.11.2013 9:24:25
dcbe4c72-6bcc-4dc6-b2d2-d3565e61fb77	dcbe4c72-6bcc-4dc6-b2d2-d3565e61fb77	Добавлена возможность выписать рецепт сроком на 3 месяца.	Загружен	27.11.2013 9:24:26
e214c339-e881-427b-9ade-62e986c1d14f	e214c339-e881-427b-9ade-62e986c1d14f	Описание для файла обновления не задано	Загружен	27.11.2013 9:24:25
ec88cbdc-1f5f-48fc-8252-057dc4ff41ab	ec88cbdc-1f5f-48fc-8252-057dc4ff41ab	Исправлена ошибка вывода льготы при открытии рецепта на редактирование. Изменена проверка врачей при сохранении рецепта со статусом "испорчен".	Загружен	27.11.2013 9:24:28
eee9d288-f030-4c19-9021-4e6468c890a6	eee9d288-f030-4c19-9021-4e6468c890a6	Добавлен отчет "Мониторинг по группам населения и категориям заболеваний"	Загружен	27.11.2013 9:24:25

Рисунок 61. Журнал «История проведения обновлений»

2.2.2.4 Журнал протоколов

Журнал протоколов – средство контроля над операциями экспорта/импорта. Журнал доступен в подпункте меню «Операции» → «Журналирование» → «Журнал протоколов». Диалог позволяет просмотреть полную историю выполненных протоколов (Рисунок 62) и перечень возникших в ходе операций ошибок (Рисунок 63).

Идентификатор операции	Имя протокола	Версия протокола
9dabf611-015c-4b21-8005f...	Выгрузка посещений врача	1.2
b62b368f-1647-4963-aaf2-7...	Выгрузка карт ПФО	1.0
ba0c05de-3e69-40f8-95c9-...	Выгрузка реестра выписанных реце...	2.1
bf047ba0-3e34-41d7-9005-...	Выгрузка данных о направлениях на ...	2.7.3
3358a2fd-d65d-42fe-a6c6-e...	Выгрузка данных о направлениях, го...	2.8.9993
61258c9d-05f3-4834-9e5d-...	Выгрузка медицинской истории	2.2.3
a4e8358d-5b6e-4497-952e-...	Загрузка рецептов аптек	2.3
a4e8358d-5b6e-4497-952e-...	Загрузка рецептов аптек	2.3
72af55ae-96a0-4434-9c15-...	Приказ N79\2. Создание реестра ок...	1.53.135
1beb2610-3c42-4da1-9cc1-...	Приказ N79\3. Выгрузка реестра в Т...	1.7.116
feada8c4-415f-4b8d-97ed-1...	Приказ N79\3. Выгрузка реестра в Т...	1.7.116
a73ba43a-64c6-49d5-b7ab-...	Приказ N79\3. Выгрузка реестра в Т...	1.7.116
dd498276-c108-4784-9dfe-...	Приказ N79\3. Выгрузка реестра в Т...	1.7.116
b5b20ee6-c112-4387-907d-...	Приказ N79\3. Выгрузка реестра в Т...	1.7.116
3c7cc4cc-8afd-4ff0-9972-1...	Приказ N79\3. Выгрузка реестра в Т...	1.7.116
5c62b939-7caf-4ddd-8a9f-3...	Приказ N79\2. Создание реестра ок...	1.53.135
0f64c17a-6b25-4acd-98a7-...	Приказ N79\3. Выгрузка реестра в Т...	1.7.116
0d874081-6d78-47c7-b3a4-...	Выгрузка реестра листов нетрудос...	2.0

Общая информация

Версия протокола 2.0

Время запуска 14.10.2014 20:03

Загружавшийся фа

Идентификатор опе 4d54a790-76e8-4642-86

Идентификатор прс 66e37267-b988-4962-bb

Имя протокола Выгрузка реестра листе

История выполнени <style type="text/css"> *

Ошибка

Пользователь Admin

Пользователь Admin

Сеанс 486339

Версия протокола

Рисунок 62. Журнал протоколов

Для просмотра перечня ошибок, возникших в ходе операций, нужно выделить требуемую запись и нажать на панели инструментов кнопку .

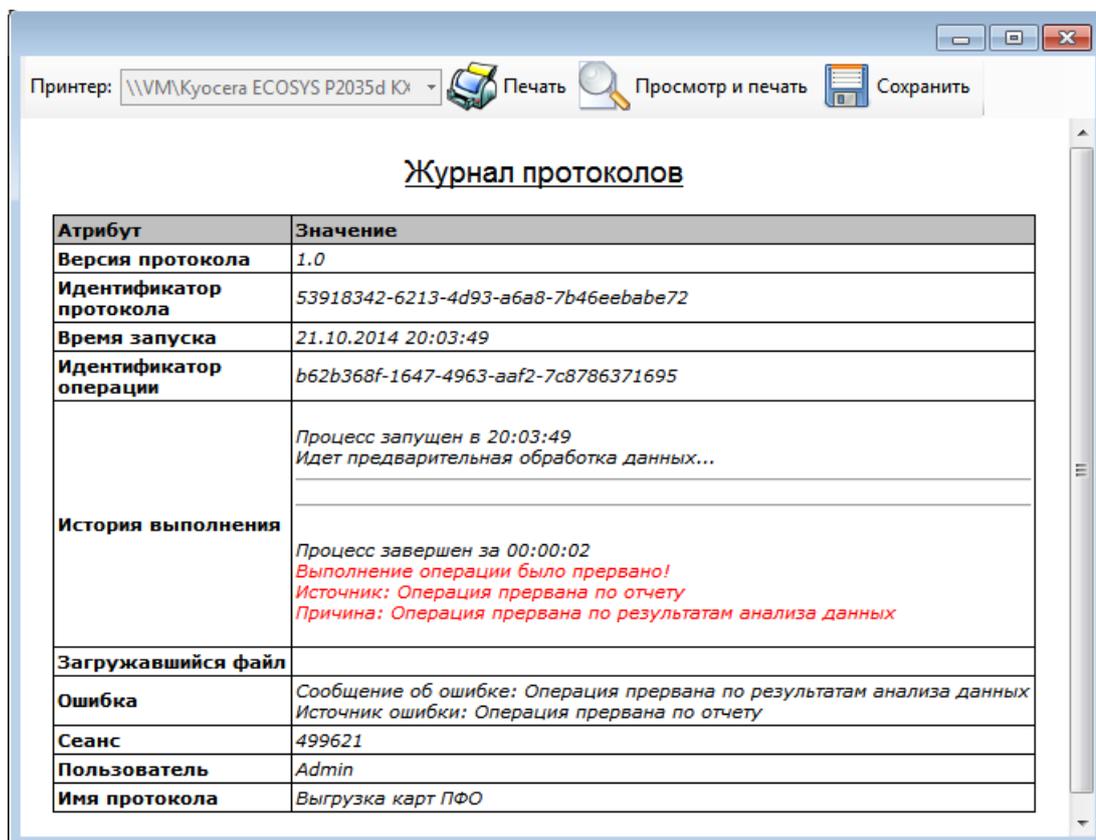


Рисунок 63. Просмотр сведений о протоколе

2.2.2.5 Журнал отчетов

Журнал отчетов – средство контроля над операциями выполнения отчетов. Журнал доступен в подпункте меню «Операции» → «Журналирование» → «Журнал отчетов». Диалог позволяет просмотреть полную историю выполненных отчетов (Рисунок 64) и полную информацию о выполненном отчете конкретным пользователем (Рисунок 65).

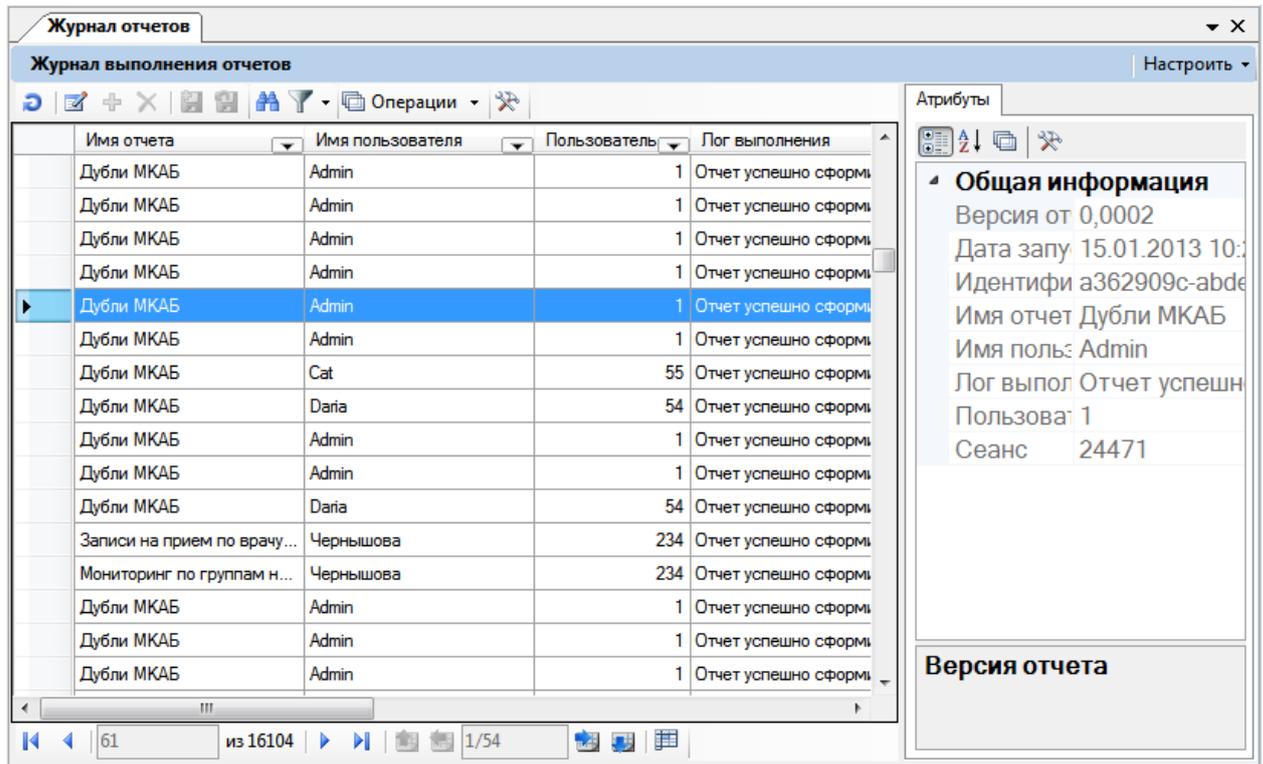


Рисунок 64. Журнал отчетов

Для просмотра полной информации о выполненном отчете конкретным пользователем нужно требуемую запись выделить и нажать на панели инструментов кнопку . Откроется отчет (Рисунок 65).

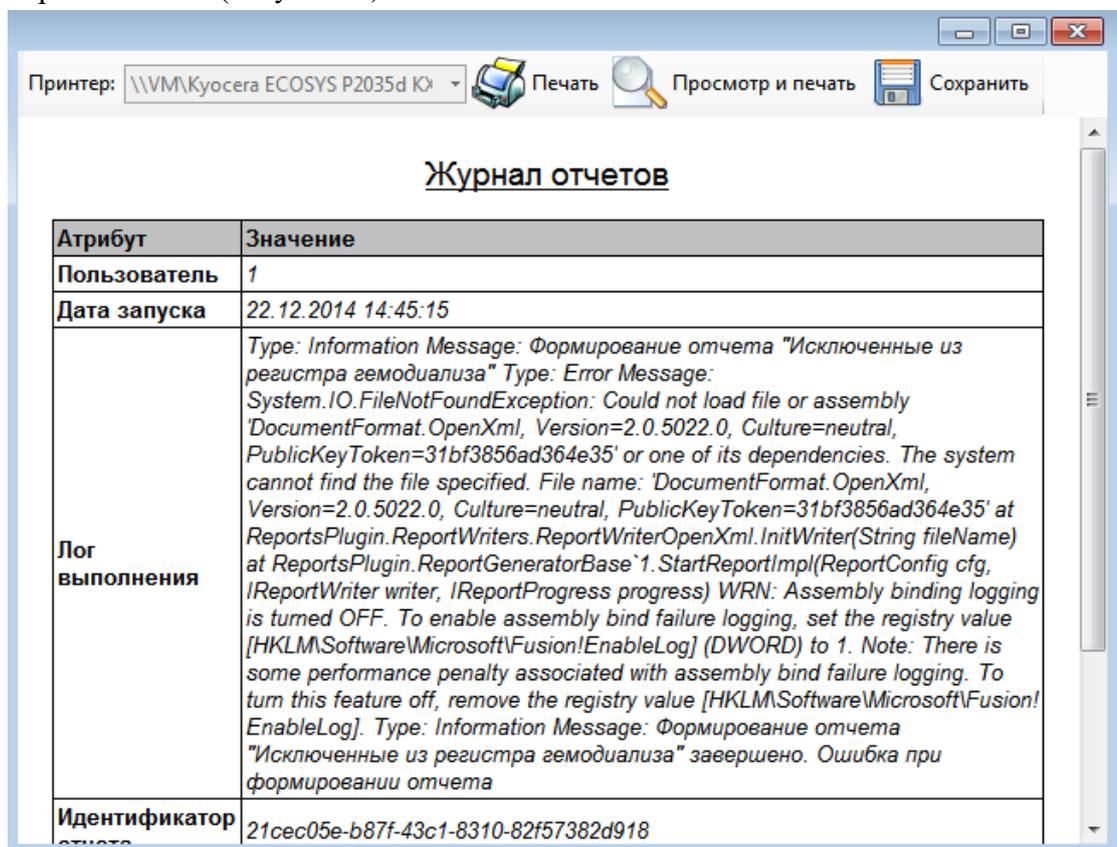


Рисунок 65. Просмотр сведений о конкретном отчете

2.2.3 Резервное копирование базы данных

В режиме администрирования Программы администратору доступна для просмотра и редактирования информация, содержащаяся в БД в виде таблиц. Полный доступ к управлению БД достигается средствами Microsoft SQL Server Management Studio.

Доступ для копирования базы данных осуществляется в подпункте меню «Операции» → «Резервное копирование базы данных» (Рисунок 66).

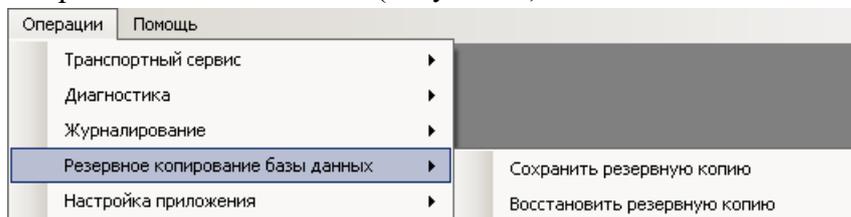


Рисунок 66. Окно выбора операции

2.2.3.1 Создание резервной копии базы данных

Для создания резервной копии базы данных необходимо выбрать подпункт меню «Операции» → «Резервное копирование базы данных» → «Сохранить резервную копию». Далее необходимо выбрать файл, в который будет сохраняться резервная копия базы данных (Рисунок 67).

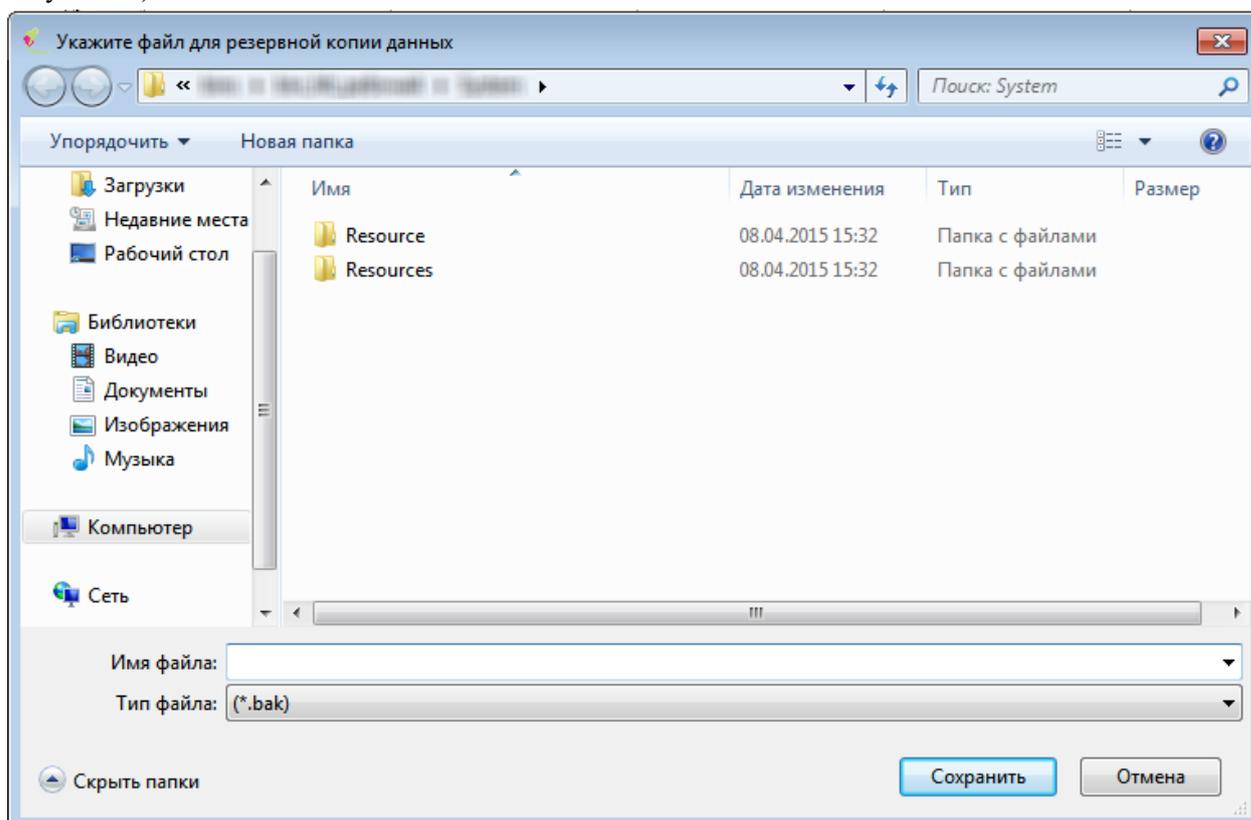


Рисунок 67. Диалог выбора папки и файла для резервной копии БД

Для этого в появившемся окне надо выбрать путь и ввести название файла. По кнопке «Сохранить» произведется сохранение резервной копии данных в файл, указанный в диалоге.

2.2.3.2 Восстановление базы данных из резервной копии

Для восстановления резервной копии базы данных необходимо выбрать подпункт меню «Операции» → «Резервное копирование базы данных» → «Восстановить резервную копию». Восстановление производится в полученном диалоге, представленном ниже (Рисунок 68).

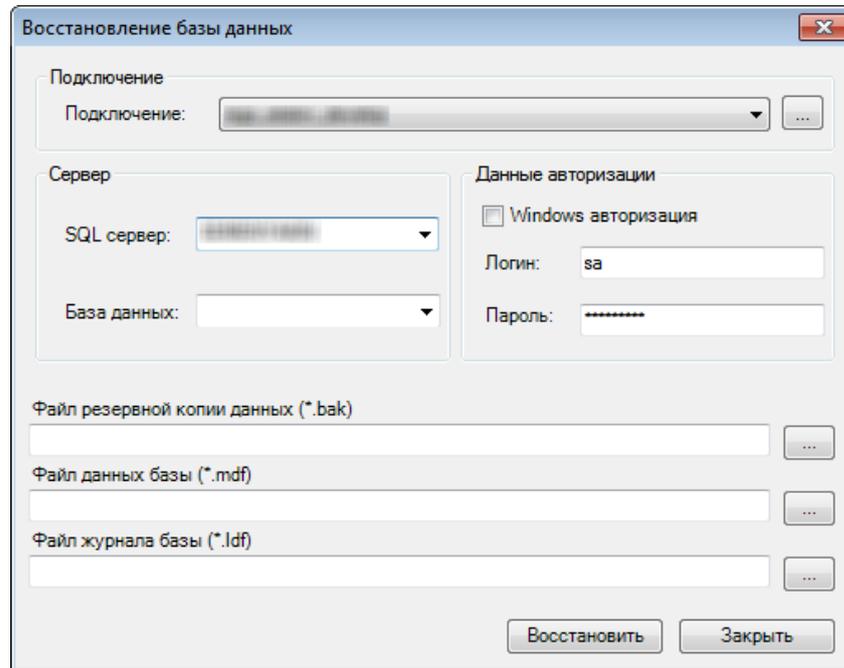


Рисунок 68. Диалог восстановления БД

В диалоге необходимо выбрать существующее подключение либо настроить его вручную: ввести название сервера баз данных, на котором будет производиться восстановление копии базы, название новой базы данных, ввести имя пользователя и пароль на сервере базы данных или выбрать пункт «Windows авторизация», если сервер баз данных поддерживает данный тип авторизации.

Следующим шагом необходимо выбрать файл, который содержит в себе резервную копию базы данных, а также указать расположение файлов, которые будут использоваться в качестве рабочей базы данных. По кнопке «Восстановить» начнется процесс восстановления данных.

2.2.4 Настройка приложения

Доступ для настройки приложения осуществляется в подпункте меню «Операции» → «Настройка приложения» (Рисунок 69).

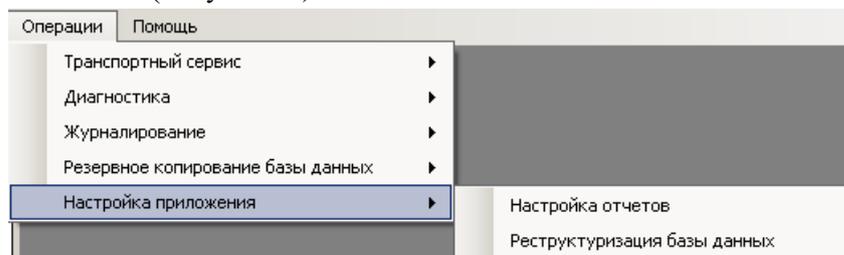


Рисунок 69. Окно выбора операции

2.2.5 Реструктуризация базы данных

Реструктуризация – регламентное, автоматическое изменение структуры базы данных, проводимое под управлением платформы, в соответствии с файлом реструктуризации. Файл реструктуризации поставляется организацией, занимающейся сопровождением ПК.

Пункт «Операции» → «Настройка приложения» → «Реструктуризация базы данных» позволяет выполнить реструктуризацию базы, к которой в данный момент подключен Администратор.

При вызове данного пункта открывается диалог, представленный на рисунке (Рисунок 70).

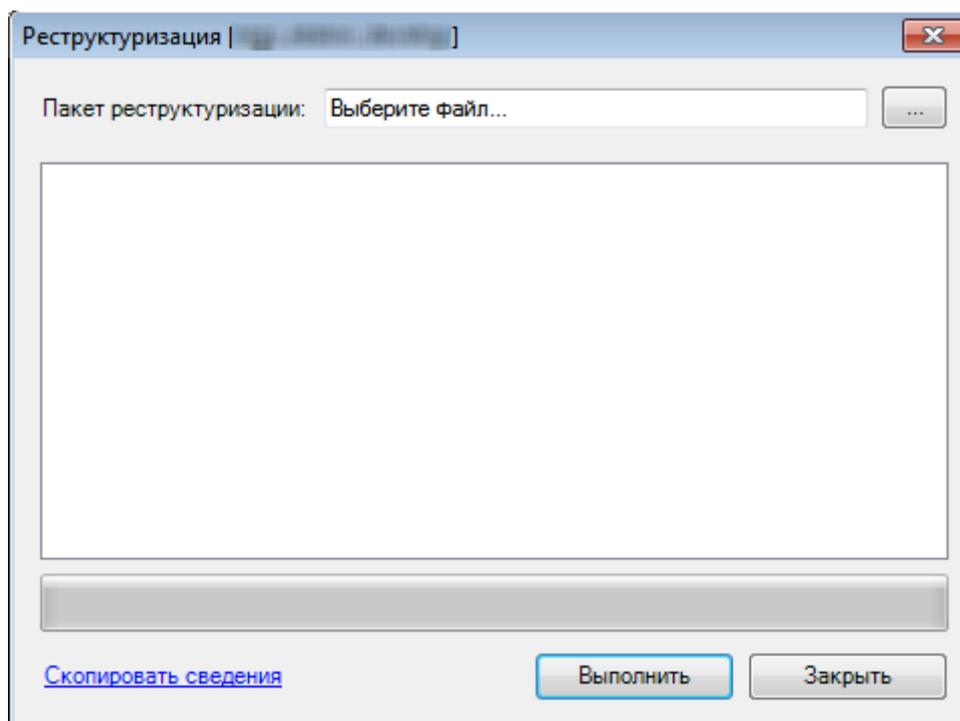


Рисунок 70. Диалог выполнения реструктуризации

Поле «Пакет реструктуризации» данной формы позволяет выбрать все возможные реструктуризации, которые могут быть проведены на данной базе. Чтобы выполнить реструктуризацию необходимо выбрать нужный пакет и нажать кнопку «Выполнить». Кнопка «Закрыть» позволяет выйти без проведения реструктуризации.

Если не выбран пакет реструктуризации, Программа выдаст предупреждение (Рисунок 71).

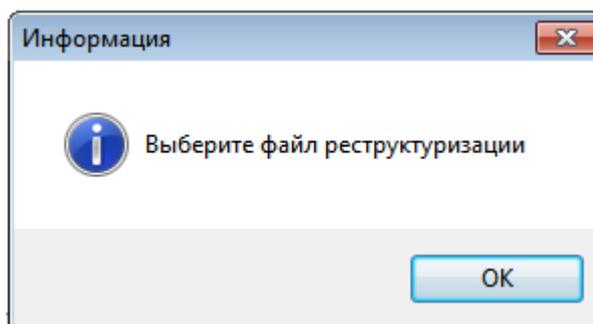


Рисунок 71. Уведомление о невыбранном пакете реструктуризации

По окончании реструктуризации пользователю будет предоставлен отчет о завершении реструктуризации (Рисунок 72). В том же окне будет собрана служебная информация.

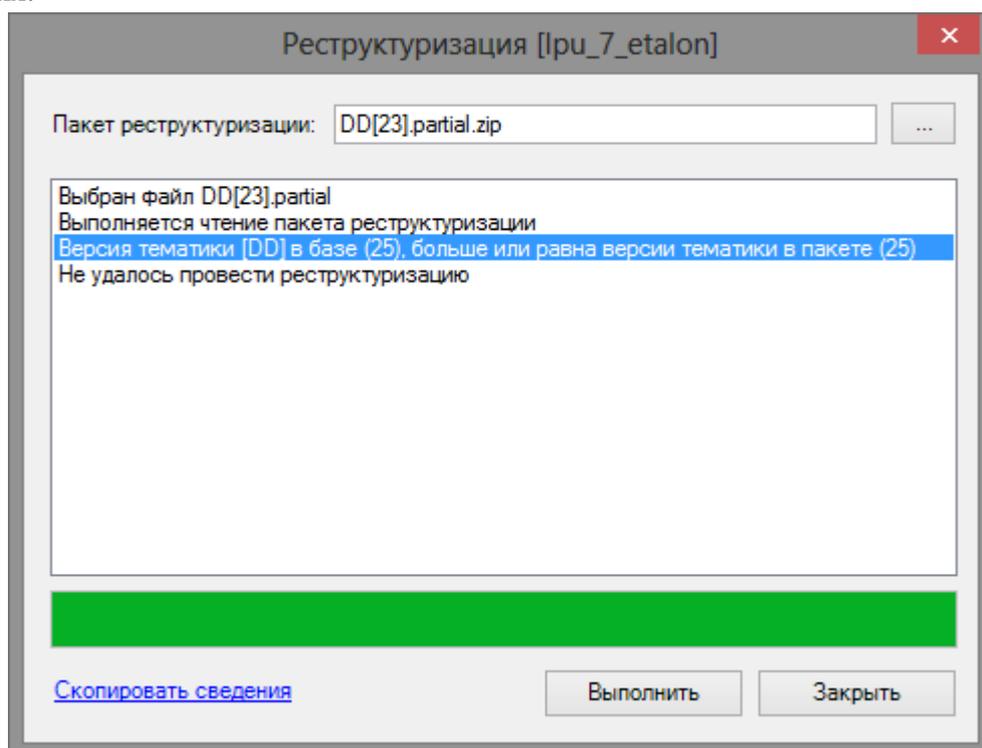


Рисунок 72. Отчет о проведении реструктуризации

2.2.5.1 Настройка формирования отчетов

Выбор приложения, используемого для формирования отчета, выполняется в диалоге «Операции» → «Настройка приложения» → «Настройка отчетов». Система формирования отчетов может использовать либо Microsoft Excel, либо OpenOffice Calc (Рисунок 73). «Глобальная» настройка устанавливает используемое приложение для всех клиентов данной БД. «Локальная» настройка устанавливает приложение для текущего компьютера.

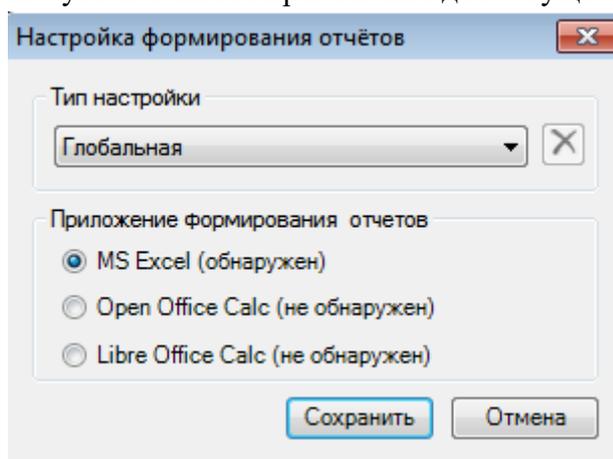


Рисунок 73. Настройка формирования отчетов

2.3 Методы и сценарии установки программного обеспечения рабочих станций и серверов

Перед установкой «ТМ:ЦОД.МЭК» необходимо выполнить установку следующих компонент системного ПО (Таблица 3):

1. Windows Installer 3.1;
2. MS .Net Framework 3.5;
3. MDAC 2.8, или выше (если используется ОС Windows 2000);
4. MS SQL Server 2008 или выше;
5. MS SQLXML 4.0 SP1 (При установке MS SQLXML 4.0 SP1 оставьте вариант установки по умолчанию).

Все приложения и командные файлы, необходимые для установки дополнительных системных компонент, находятся на компакт-диске в папке PreInstall и в папке mssql.

Таблица 3 - Перечень системных компонент на установочном диске

Наименование компоненты	Каталог на установочном диске	Имя установочного файла
Windows Installer 3.1	\PreInstall\Windows Installer 3.1	WindowsInstaller-KB893803-v2-x86.exe
MS .Net Framework 2.0	\PreInstall\NetFx2.0SP1	NetFx20SP1_x86.exe
MS SQL Express	\mssql\SQLEXP2ADV\	setup.exe
MS SQLXML 4.0 SP1	\PreInstall\MSSQLXML	sqlxml4.exe
MDAC 2.8	\PreInstall\MDAC28	MDAC_TYP.exe
MS SQLXML 4.0	\PreInstall\sqlxml4	sqlxml4.msi

Для установки компонент необходимо запустить инсталляционные файлы из соответствующих каталогов установочного диска (смотрите таблицу 2).

Для формирования отчетов требуется установленный пакет MS Office 2003 (или выше) или OpenOffice Calc 3.0 (или выше) на той машине, где планируется формировать отчетные формы.

Более подробно процесс установки ПО на серверах и рабочих станциях описан в «Руководстве по установке».

2.4 Методы проверки работоспособности рабочих станций и серверов

При запуске программы пользователю необходимо пройти авторизацию, успешное завершение данной операции свидетельствует о правильной работе сервера БД.

Проверить работоспособность текущего соединения с сервером можно, вызвав форму редактирования подключения (Рисунок 74).

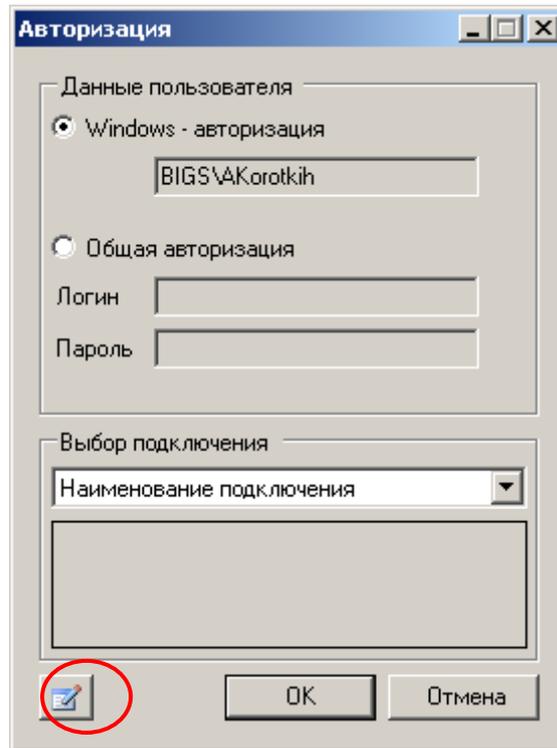


Рисунок 74. Авторизация

После чего будет показана форма с указанием наименования подключения, базы данных, к которой происходит подключение и наименование сервера (Рисунок 75).

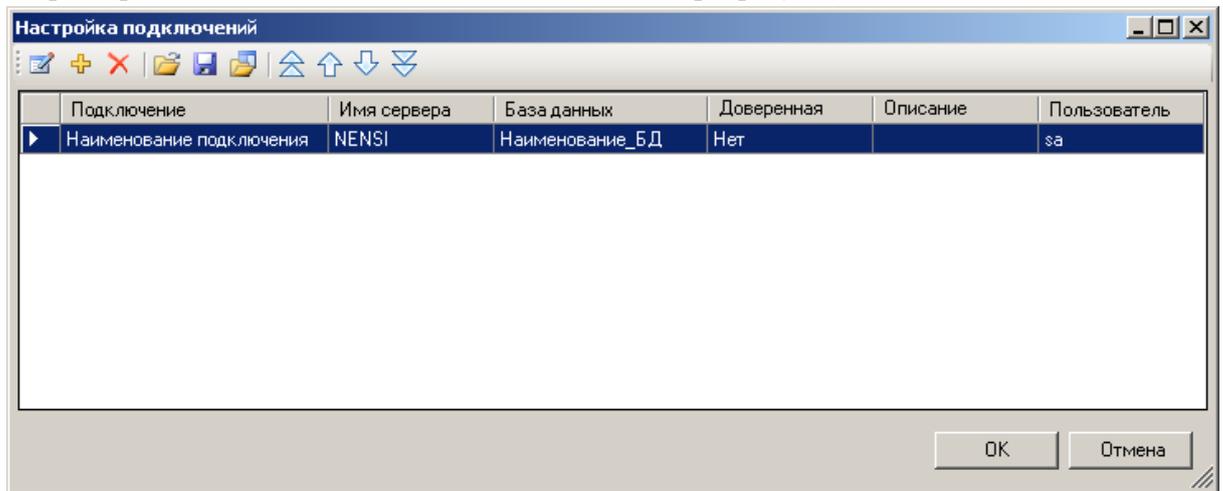


Рисунок 75. Окно «Настройка подключений»

Чтобы отредактировать параметры подключения в окне «Настройка подключений», нужно выбрать подключение из списка и нажать на кнопку  (Рисунок 76) Появится окно «Настройка подключения» (Рисунок 77):

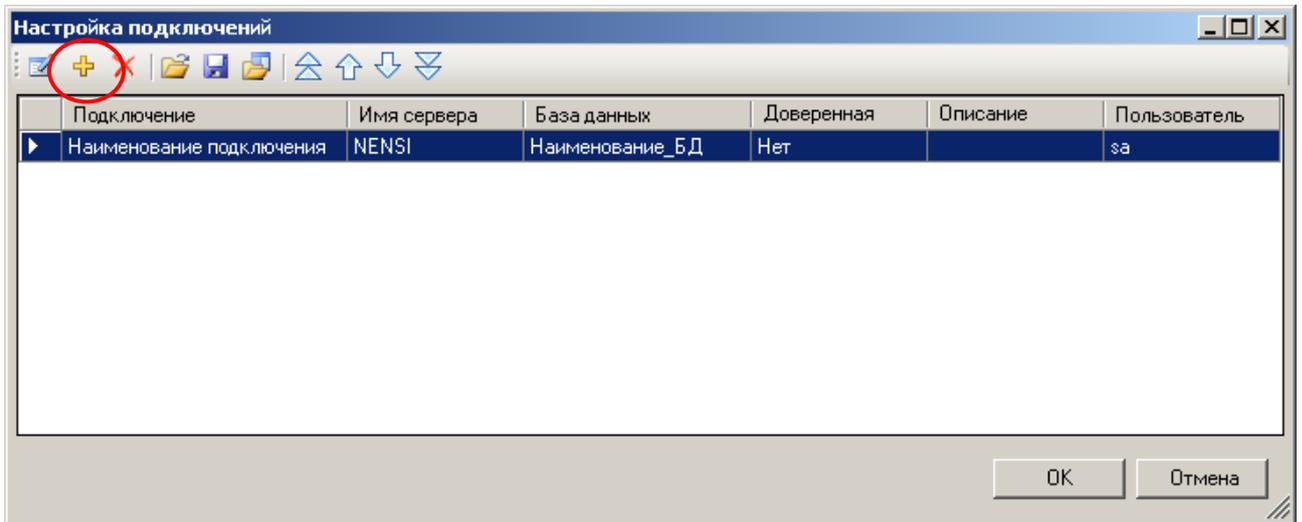


Рисунок 76. Редактирование подключений

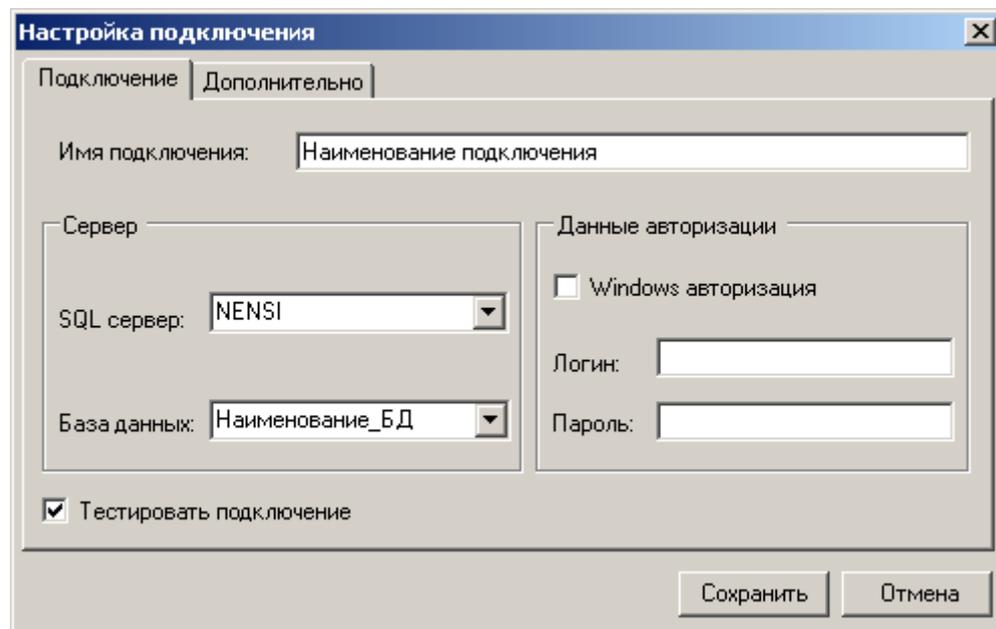
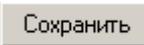


Рисунок 77. Окно «Настройка подключения»

На этой форме можно проверить работоспособность текущего подключения. Для этого необходимо установить флажок «Тестировать подключение» и нажать кнопку . Если подключение к рабочей базе данных на сервере прошло успешно, то программа сохранит заданные пользователем настройки подключения и закроет окно «Настройка подключения».

Если в момент подключения произошли неполадки в работе сервера, то будет выдано сообщение от программы (Рисунок 78).

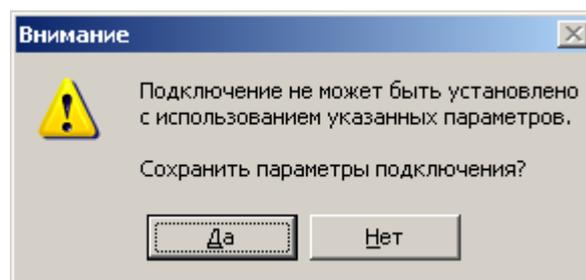


Рисунок 78. Ошибка при подключении к БД на сервере

Далее необходимо вернуться на первоначальную форму (Рисунок 74) и завершить авторизацию пользователя в системе. Если соединение с БД прошло успешно, то будет показано главное окно программы (Рисунок 79).

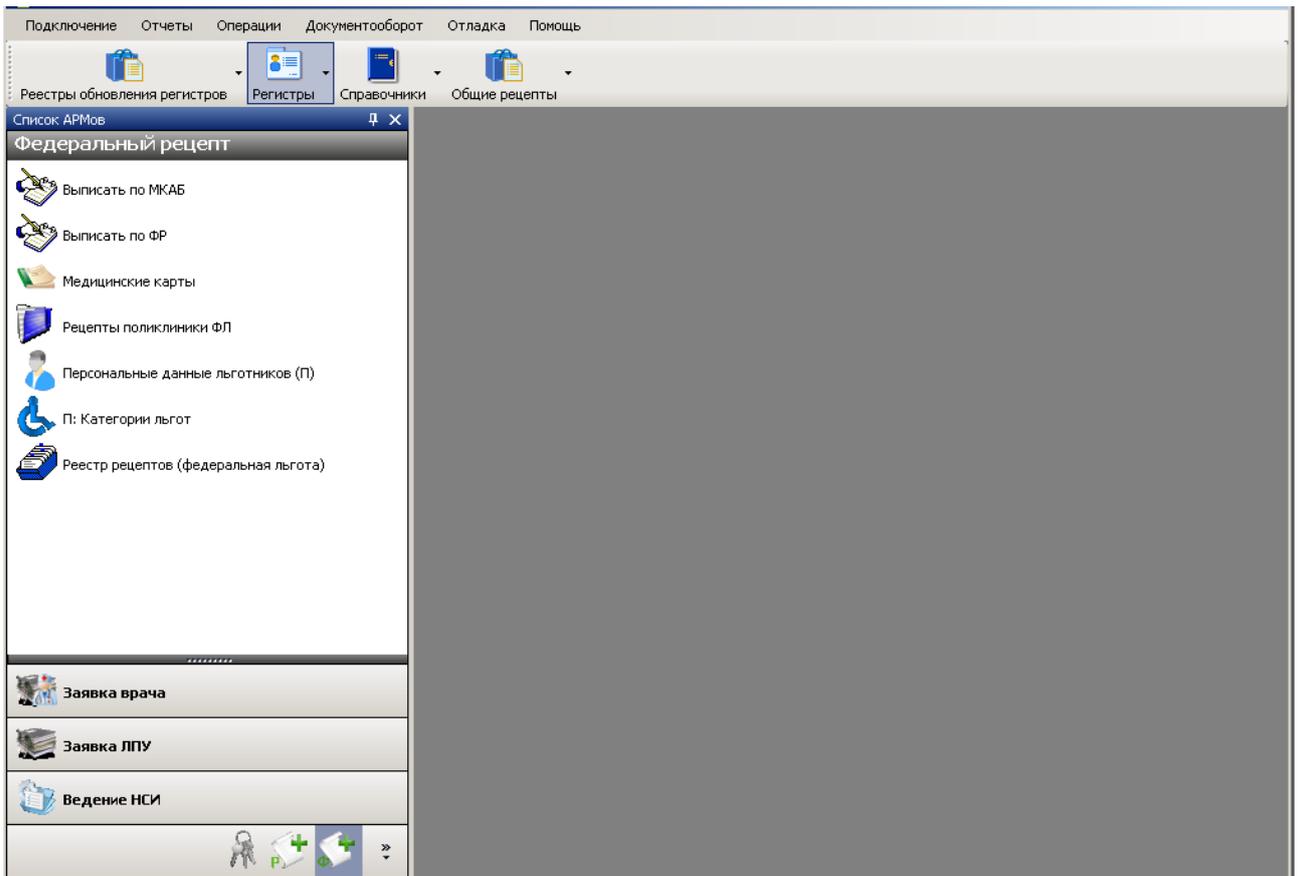


Рисунок 79. Главное окно программы

В обратном случае будет выдано сообщение о возможных причинах ошибки (Рисунок 80).

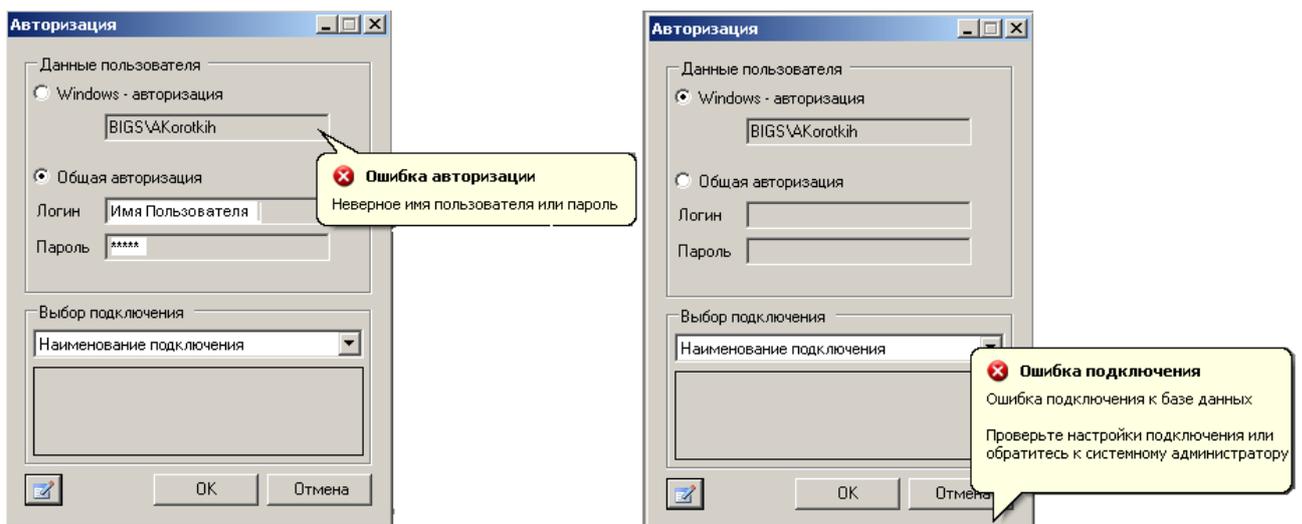


Рисунок 80. Варианты ошибок при подключении к серверу

2.5 Методы восстановления работоспособности рабочих станций и серверов

При возникновении неполадок в работе программы (клиентской части) одним из методов восстановления ее работоспособности является завершение работы программы с последующим ее перезапуском.

Если при попытке авторизации программа выдает сообщение «Неверное имя пользователя или Пароль», то необходимо убедиться, что в БД существует пользователь с таким именем и паролем. Для этого необходимо выбрать группу «Пользователи», в которой перечислены все пользователи программы и их пароли. Здесь можно отредактировать данные конкретного пользователя или создать нового пользователя и установить ему пароль.

Причины ошибок в работе программы могут быть связаны с повреждением рабочей базы данных, находящейся на сервере. Для устранения неполадок такого рода необходимо заново восстановить файл с базой данных. Необходимо выбрать пункт меню «Операции – Резервное копирование базы данных – Восстановить резервную копию».

Если перезапуск программы и восстановление БД не исправили ошибок в работе «ТМ:ЦОД.МЭК», то необходимо удалить данный программный продукт со своего компьютера и установить программу заново.

2.6 Методы проверки работоспособности БД

Первым шагом проверки работоспособности БД является авторизация пользователей (более подробно этот процесс описан в п. 2.4 текущего документа). Успешное завершение данной операции указывает на корректную работу БД. Подтверждением успешной авторизации является отсутствие сообщений об ошибках в ходе ее выполнения и появление главного окна программы, из которого можно осуществить доступ к различным формам документов (данные, отображаемые в этих формах, выбираются из БД).

Для проверки того, что данные из БД отображаются в таблицах правильно, необходимо открыть любую форму программы. Убедиться, что при открытии формы не произошло ошибок, а также что данные, отображаемые в таблице корректны.

2.7 Методы поддержания целостности БД

«ТМ:ЦОД.МЭК» обеспечивает поддержание целостности структуры БД. Для предотвращения возможных ошибок, связанных с потерей целостности структуры базы данных, рекомендуется выполнять ежедневную операцию резервирования БД. Из данного файла с резервной копией БД (файл с расширением .bak) можно в дальнейшем выполнить ее восстановление, что поможет устранить ошибки при работе программы с базой данных.

Создать резервную копию можно с помощью пункта меню «Операции – Резервное копирование базы данных – Создать резервную копию».

2.8 Методы поддержания безопасности БД

Безопасность базы данных обеспечивается 2-мя средствами:

1. Подключением клиентских приложений к серверу под ограниченной учетной записью;
2. регулярным резервированием БД (создание бэкапа или файла с расширением bak).

Для создания ограниченной учетной записи пользователя необходимо подключиться к серверу с помощью Microsoft SQL Server Management Studio под привилегированным пользователем (его логин как правило «sa») (Рисунок 81).

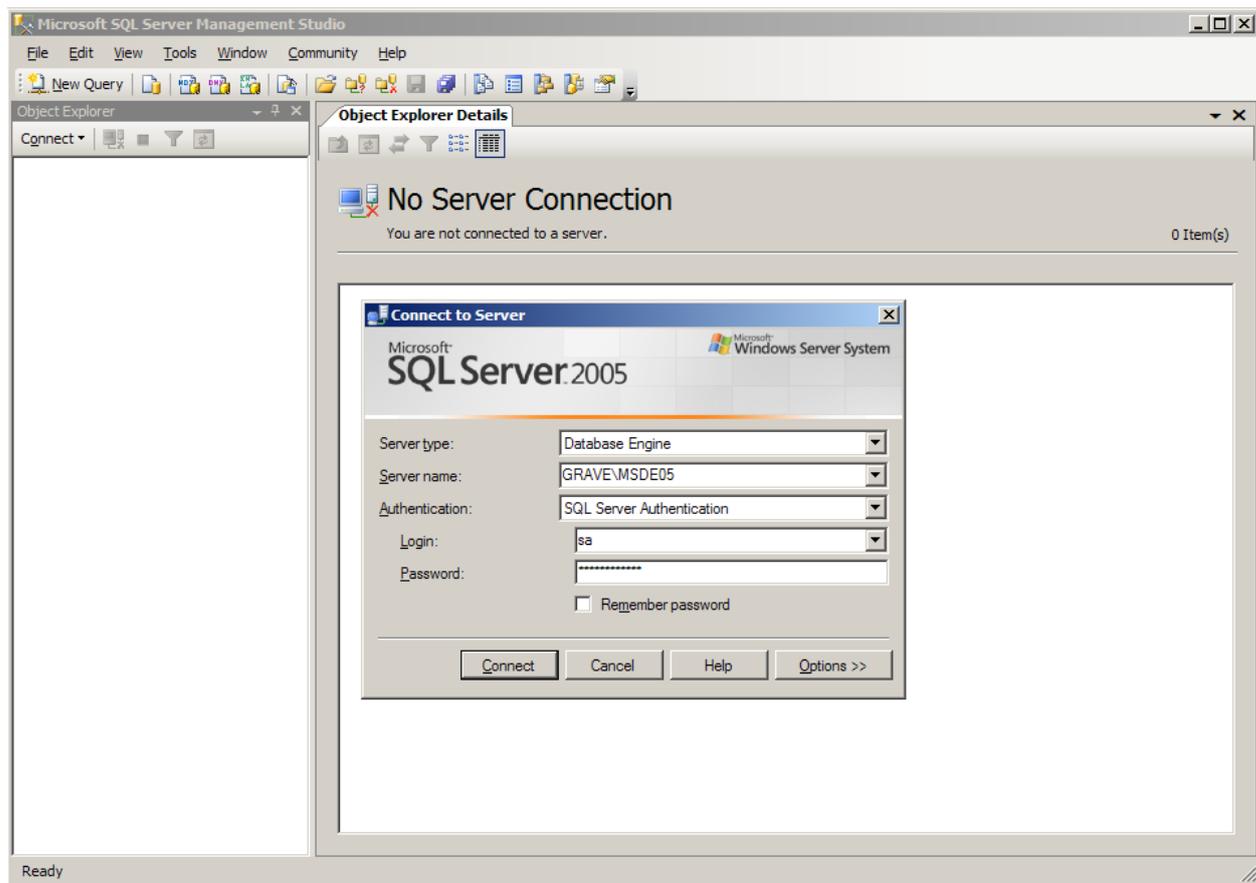


Рисунок 81. Подключение к серверу

В разделе Security\Logins создать логин пользователя (Рисунок 82).

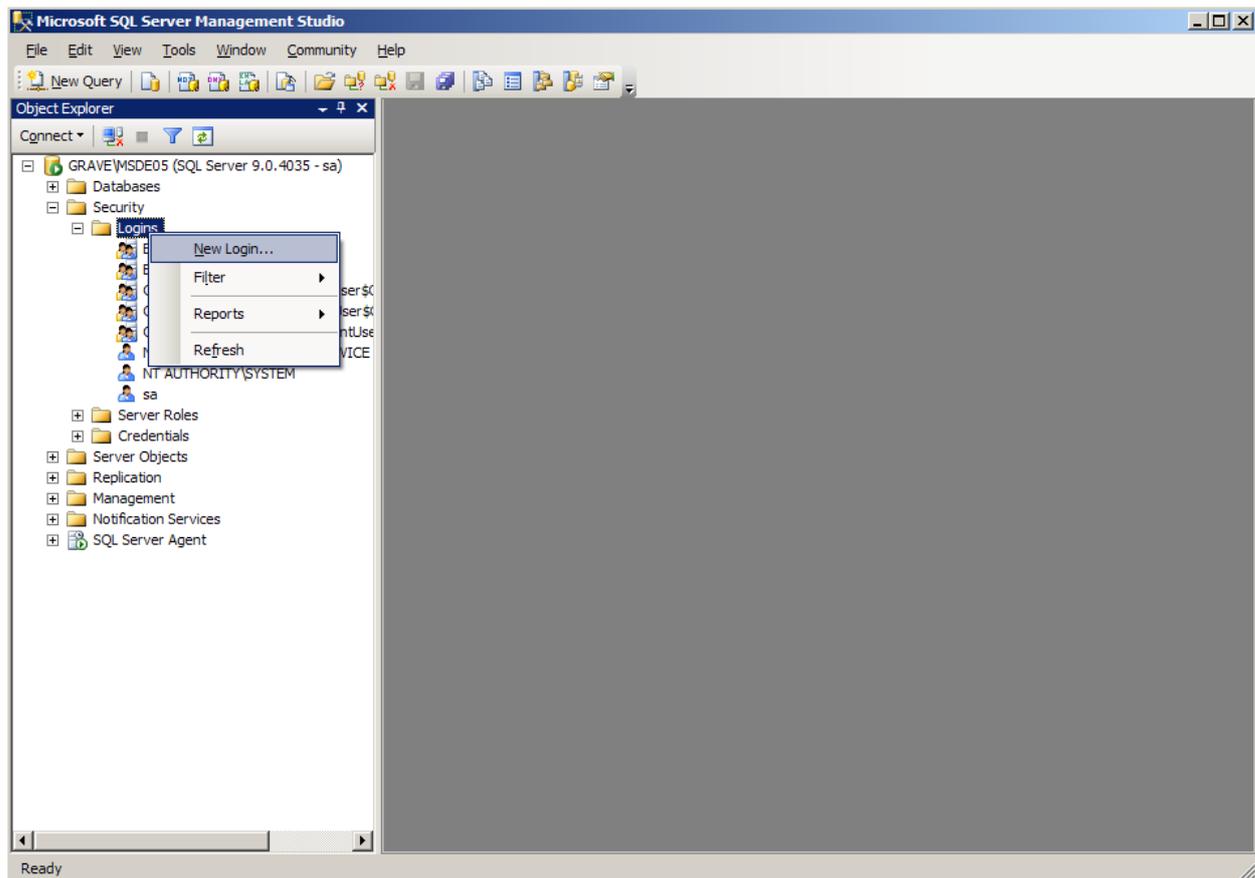


Рисунок 82. Создание логина пользователя

В открывшемся окне, в первом разделе «General» необходимо:

1. указать имя пользователя (в примере «client»);
2. указать тип авторизации «Sql Server authentication»;
3. ввести и подтвердить пароль;
4. снять флажок «User must change password at next login» (Рисунок 83).

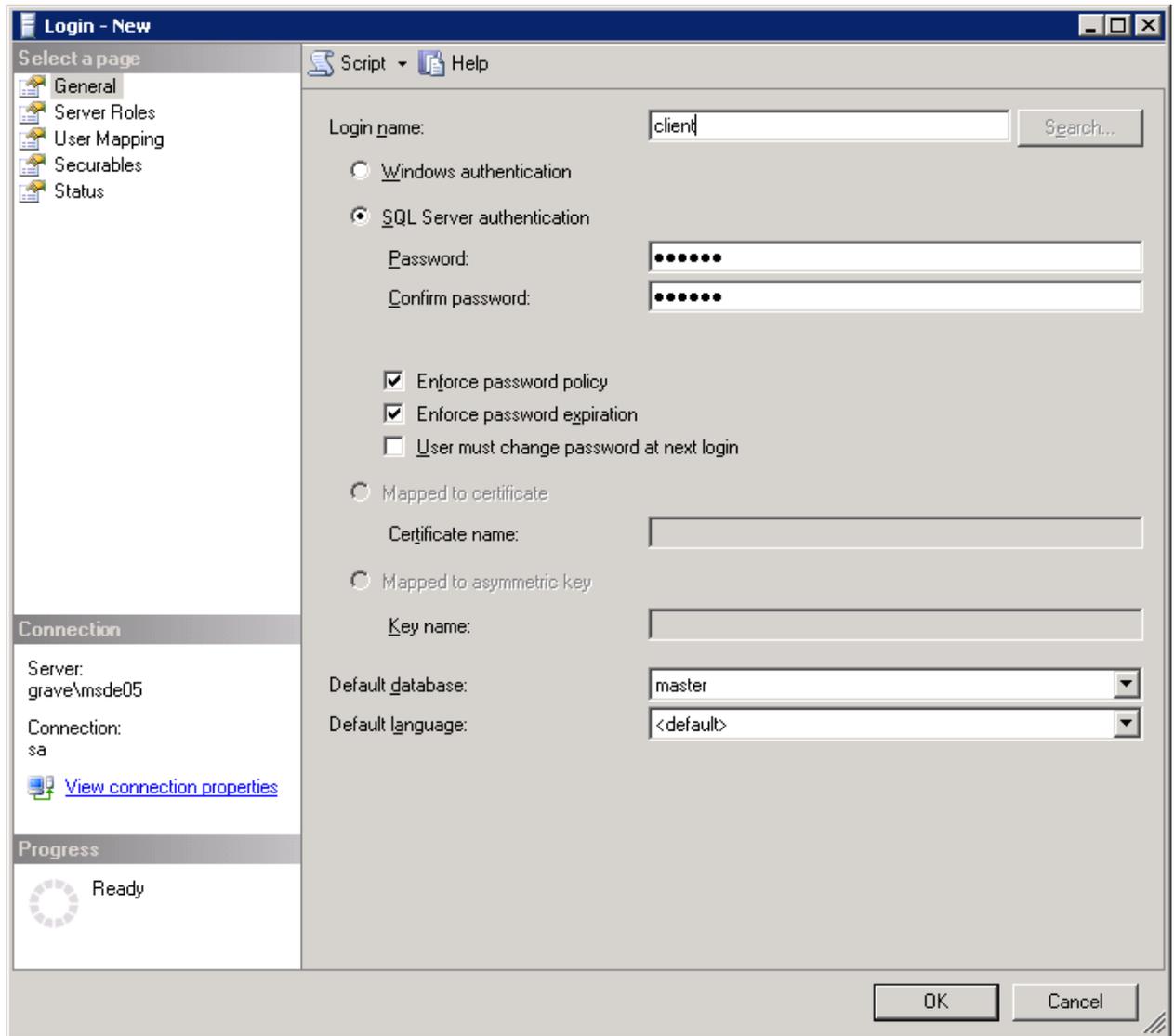


Рисунок 83. Вкладка «General»

На вкладке «User Mapping» необходимо отметить флажком базу данных, к которой пользователь должен иметь доступ (в примере это «oms_db_hlt»). В нижней части окна установить флажки для значений «public» и «db_owner» (Рисунок 84).

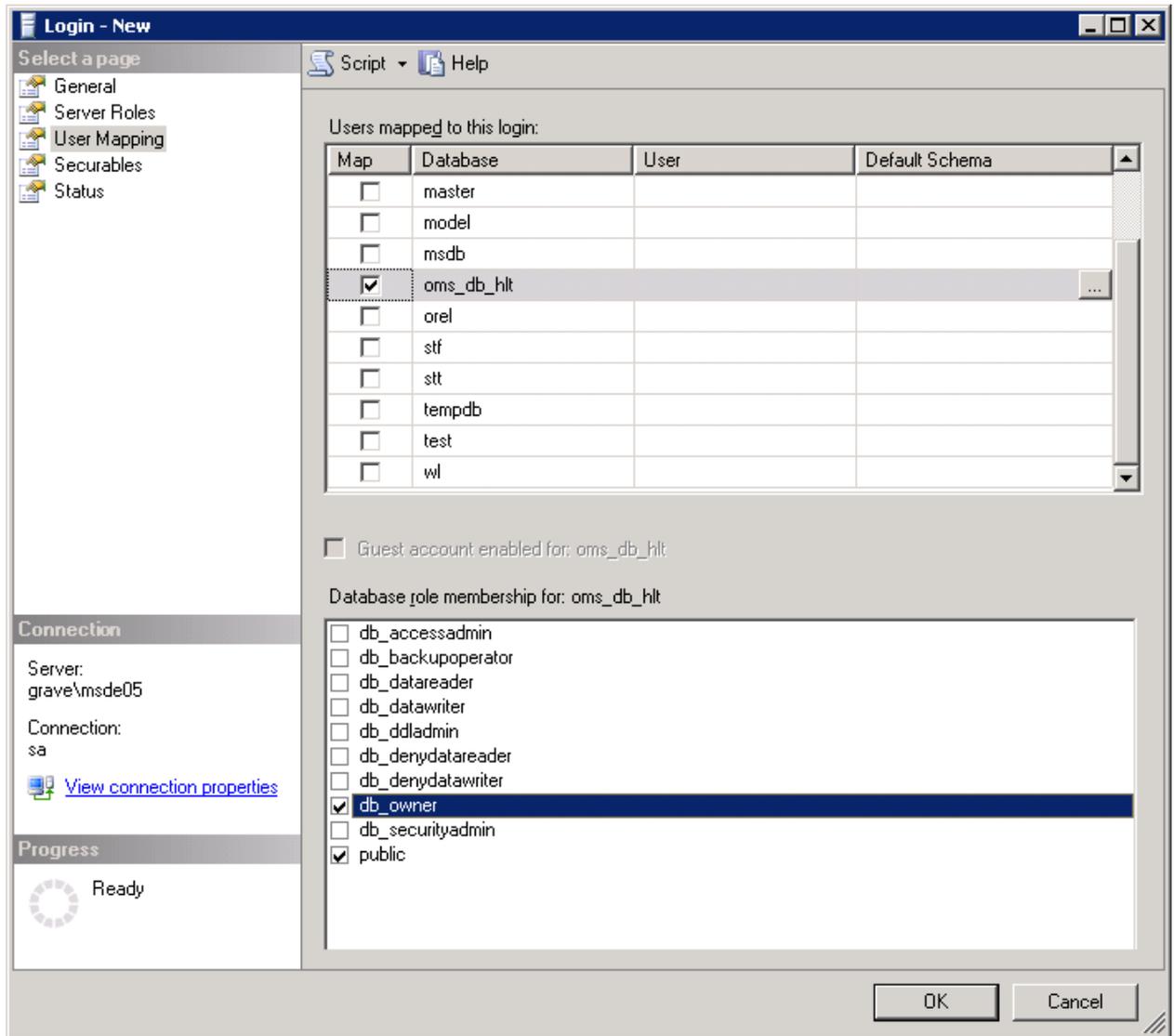


Рисунок 84. Вкладка «User Mapping»

На последней вкладке «Status» необходимо убедиться, что учетная запись включена (Рисунок 85).

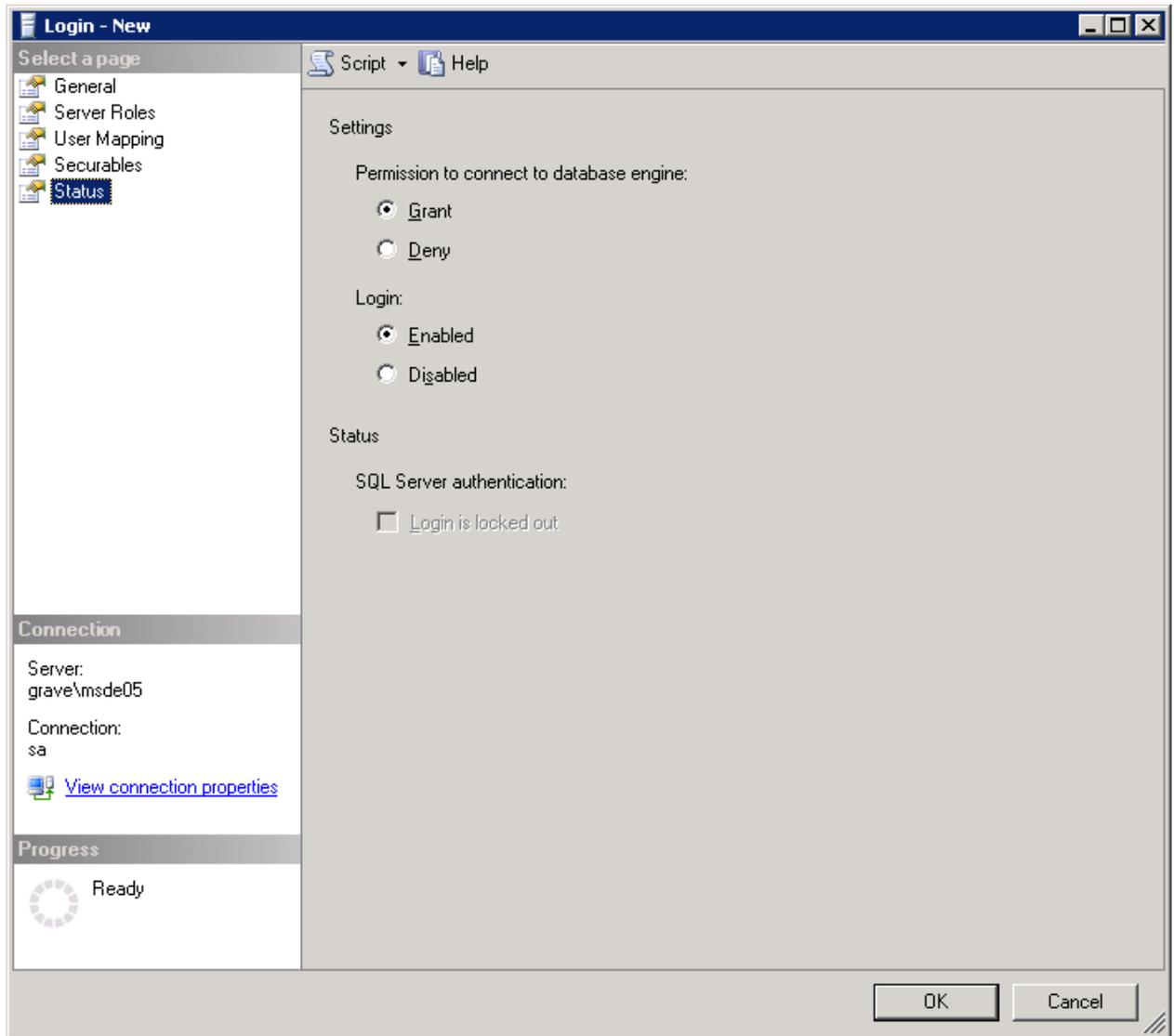


Рисунок 85. Вкладка «Status»

После нажатия кнопки «ОК» пользователь будет создан. Для проверки работоспособности созданной учетной записи необходимо выполнить переподключение к серверу.

Создать резервную копию БД (бэкап базы) можно следующими способами:

1. средствами Microsoft SQL Server Management Studio;
2. из приложения программы;
3. встроенными средствами резервного копирования.

Для создания резервной копии базы данных стандартными средствами MS SQL Server необходимо выполнить следующую последовательность действий:

1. выберите пункт меню «Task» → «Back Up...» в Microsoft SQL Server Management Studio (Рисунок 86).

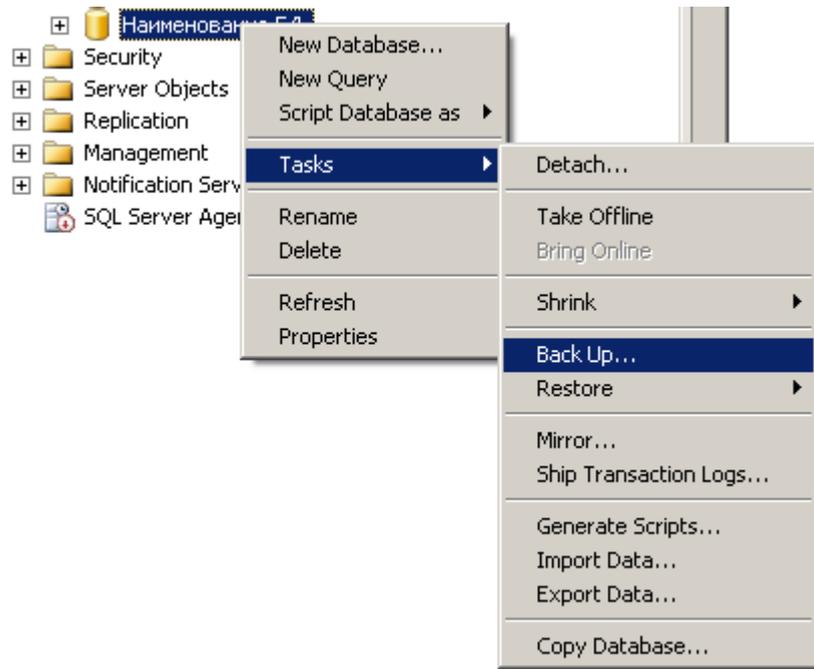


Рисунок 86. Вызов окна для создания бэкапа базы

В появившемся окне «Back Up Database» укажите параметры создания резервной копии, как на рисунке ниже, либо другие в случае необходимости. В поле «Destination» необходимо указать расположение создаваемого файла (Рисунок 87).

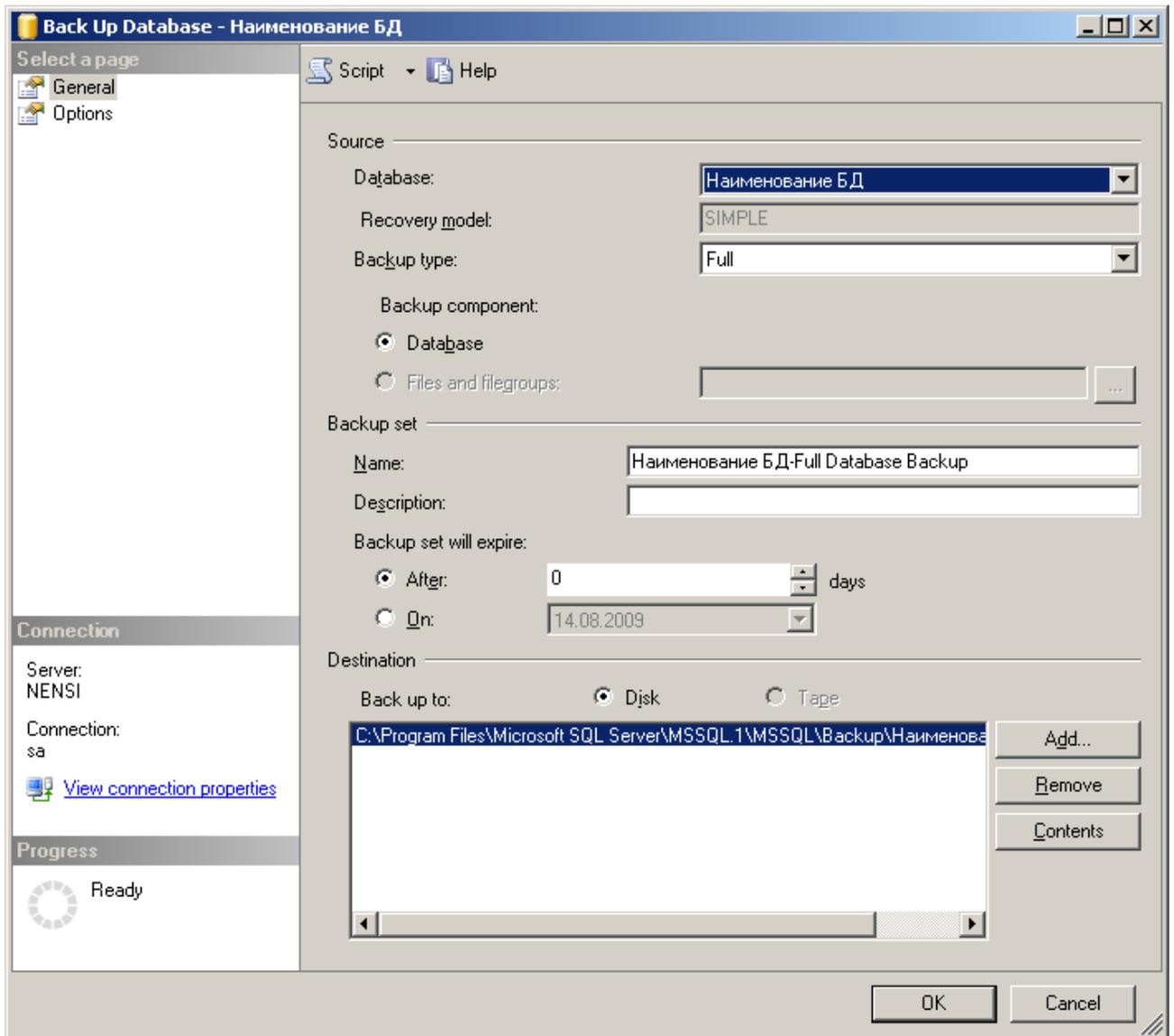


Рисунок 87. Окно «Back Up Database»

После чего нажмите кнопку . Запустится процесс создания резервной копии (Рисунок 88).

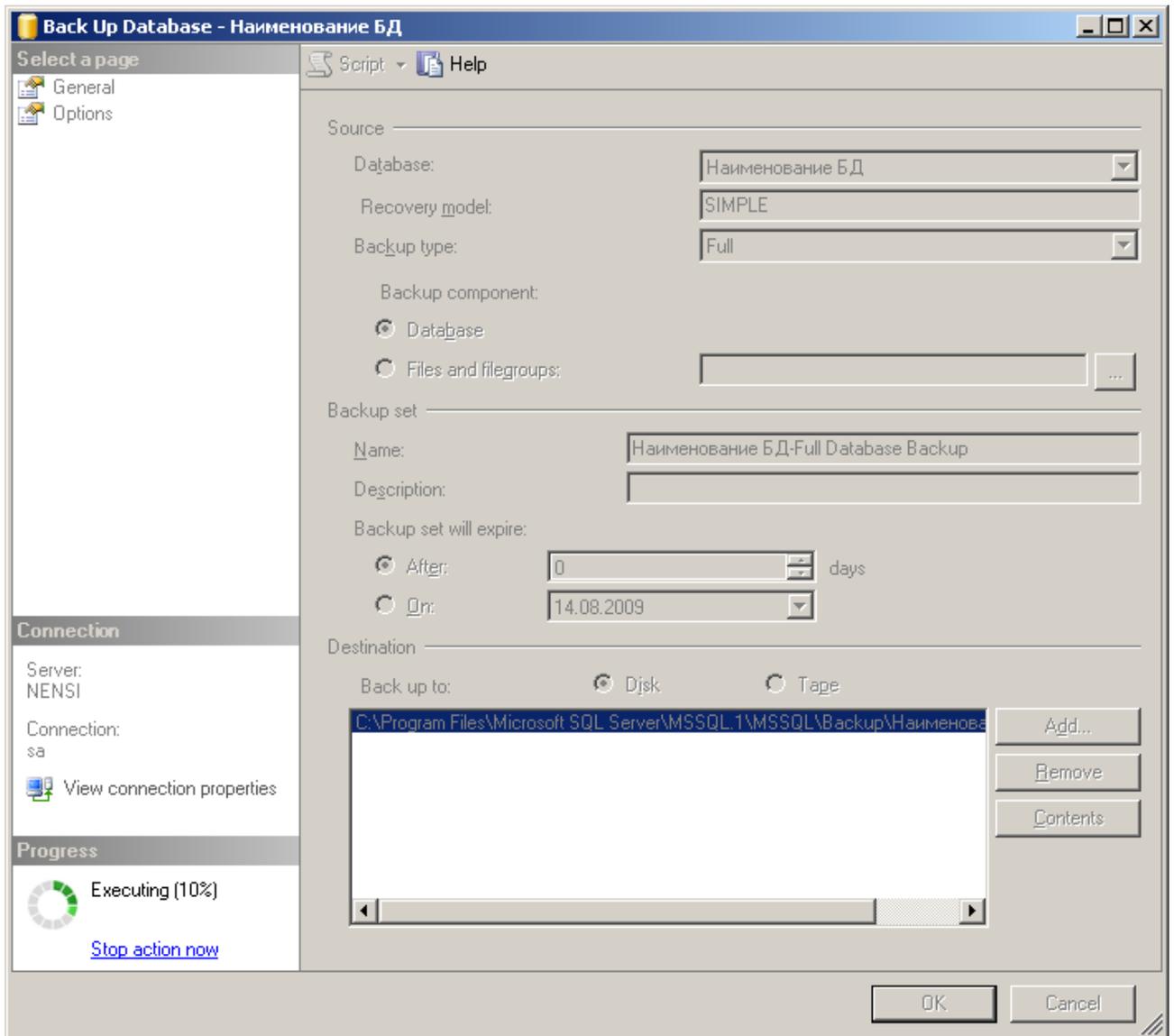


Рисунок 88. Процесс создания бэкапа базы

В случае успешного завершения операции появится окно как на рисунке ниже (Рисунок 89).

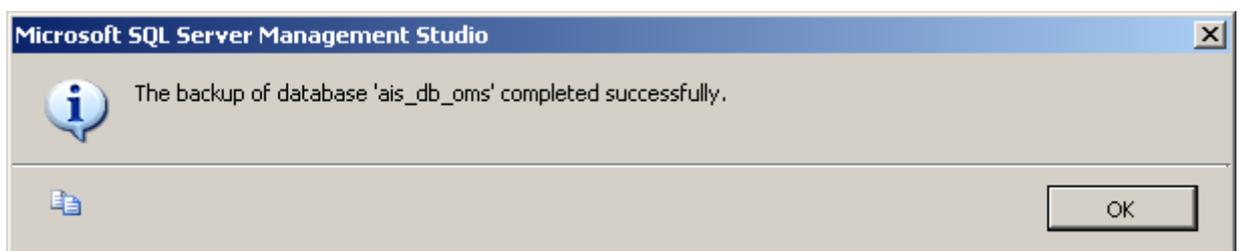


Рисунок 89. Успешное завершение операции резервирования

Подробнее о процессе создания резервной копии смотрите в документации к Microsoft SQL Server.

Процесс создания резервной копии БД через приложение «Администрирование» программы описан в п 2.5 текущего документа.

Для запуска процесса создания резервной копии базы данных встроенными средствами следует воспользоваться меню «Сценарии» → «Резервная копия БД» (Рисунок 90).

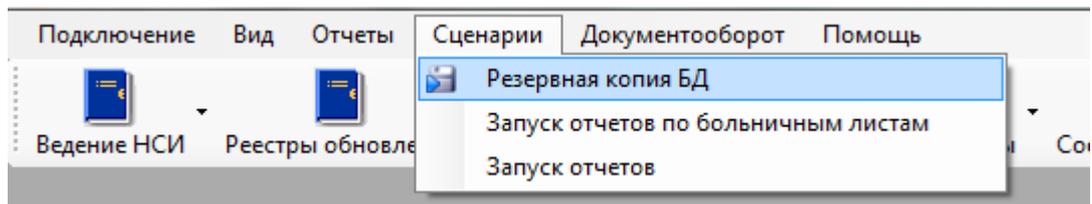


Рисунок 90. Запуск процесса создания резервной копии

После этого будет сформирован файл базы данных в папке «backups» программы.

2.9 Методы восстановления работоспособности БД

При нарушении работоспособности базы данных необходимо выполнить операцию ее восстановления из ранее созданной резервной копии (файл .bak) (более подробно об этом описано в п. 2.7 и 2.8 настоящего документа).

Приложение А Подсистема отчетов

А.1. Система описания отчетов администратором. Форма редактирования отчетов

Все отчеты подразделяются на стандартные и предопределенные. Стандартный отчет применим к текущему экземпляру документа произвольного типа, и не требует предварительного описания. Стандартный отчет формируется автоматически на основе единого для всех типов документов шаблона и описания типа документа.

Более сложные определенные отчеты предусматривают процедуру подготовки (определения) описателя отчета. Описатели определенных отчетов представляют собой экземпляры особой сущности, которые хранятся в качестве документов типа «Отчет» в БД, а также могут быть выгружены (загружены) в файлы специального формата. Создание и редактирование описателей определенных отчетов предусмотрено в приложении администрирования формой «Редактирование отчетов» (Рисунок 91 и Рисунок 92). Термин «отчет» в документе следует понимать, как описатель определенного отчета.

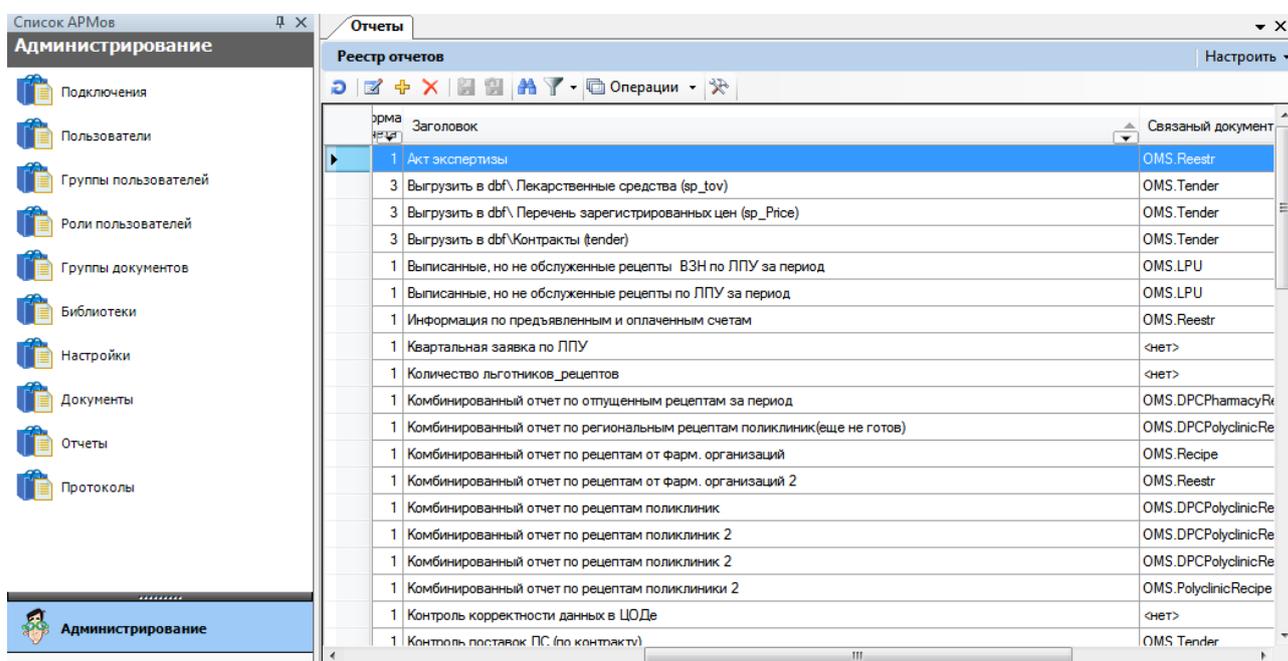


Рисунок 91. Вызов формы редактирования отчетов в приложении администрирования

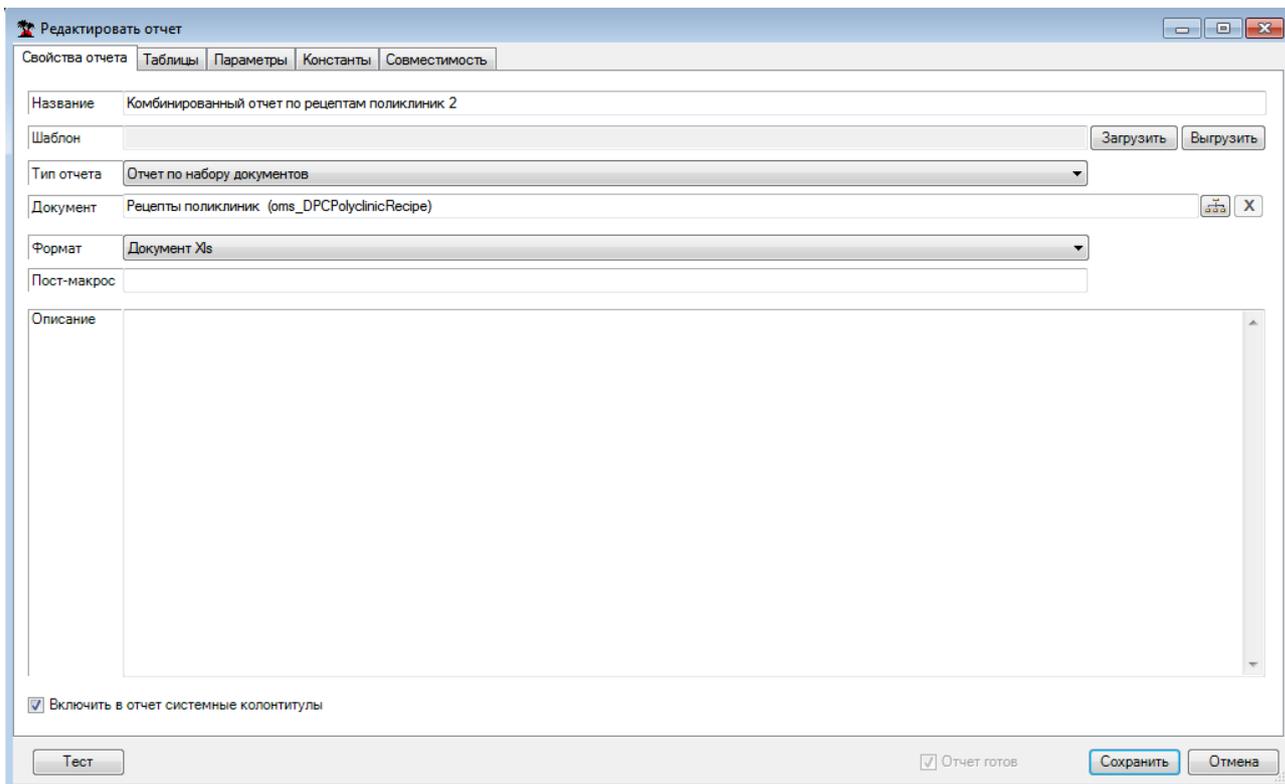


Рисунок 92. Форма редактирования отчета. Закладка «Свойства отчета»

А.1.1. Свойства отчета

Определение свойств отчета (на одноименной закладке формы редактирования отчетов) подразумевает заполнение наименования, выбор типа отчета, который определяет необходимость указания типа документа, выбор формата представления отчета, который определяет возможность загрузки и выгрузки шаблона.

А.1.1.1. Типы отчетов и их особенности

Предусмотрены три типа отчетов – «отчет по документу», «отчет по набору документов» и «произвольный отчет» (Рисунок 93).



Рисунок 93. Форма редактирования отчетов. Типы отчетов

Для первых двух обязательно задание типа документов в качестве свойства «Документ». Отчет по документу подразумевает наличие текущего экземпляра документа указанного типа, значения которого будут доступны при формировании отчета (особый параметр Document). Отчет по набору документов подразумевает инициализации текущего набора документов указанного типа с использованием фильтрации. Произвольный отчет ничего подобного не предусматривает, ограничения выборки данных задаются лишь применением интерактивных параметров.

А.1.1.2. Форматы представления отчетов

На рисунке (Рисунок 94) показан выбор формата отчета из предлагаемого списка. Формат «Excel» подразумевает необходимость подготовки файла-шаблона с использованием специальной разметки и загрузки его в описатель отчета по кнопке «Загрузить». По кнопке «Выгрузить» можно сохранить (Рисунок 95) шаблон из описателя в файл соответствующего формата (затем отредактировать его и снова загрузить).

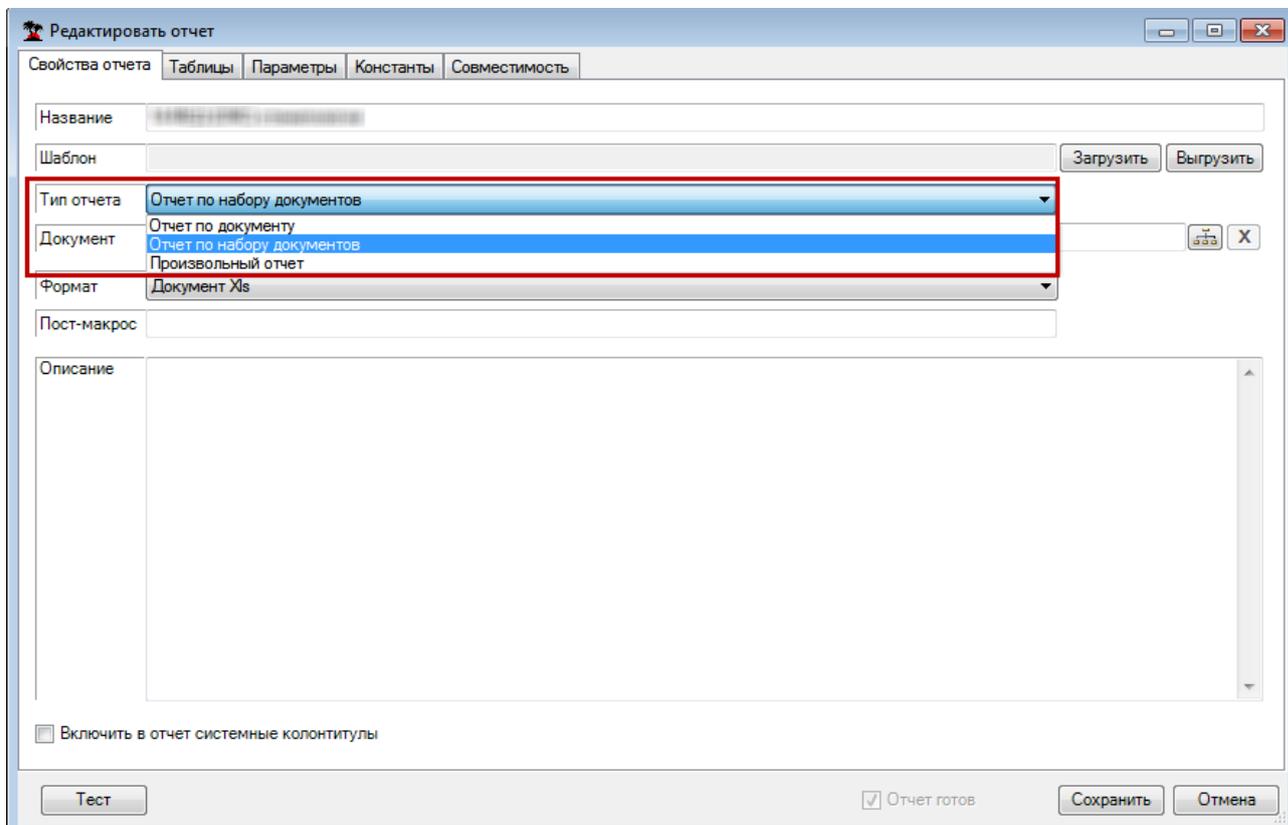


Рисунок 94. Выбор формата представления отчета

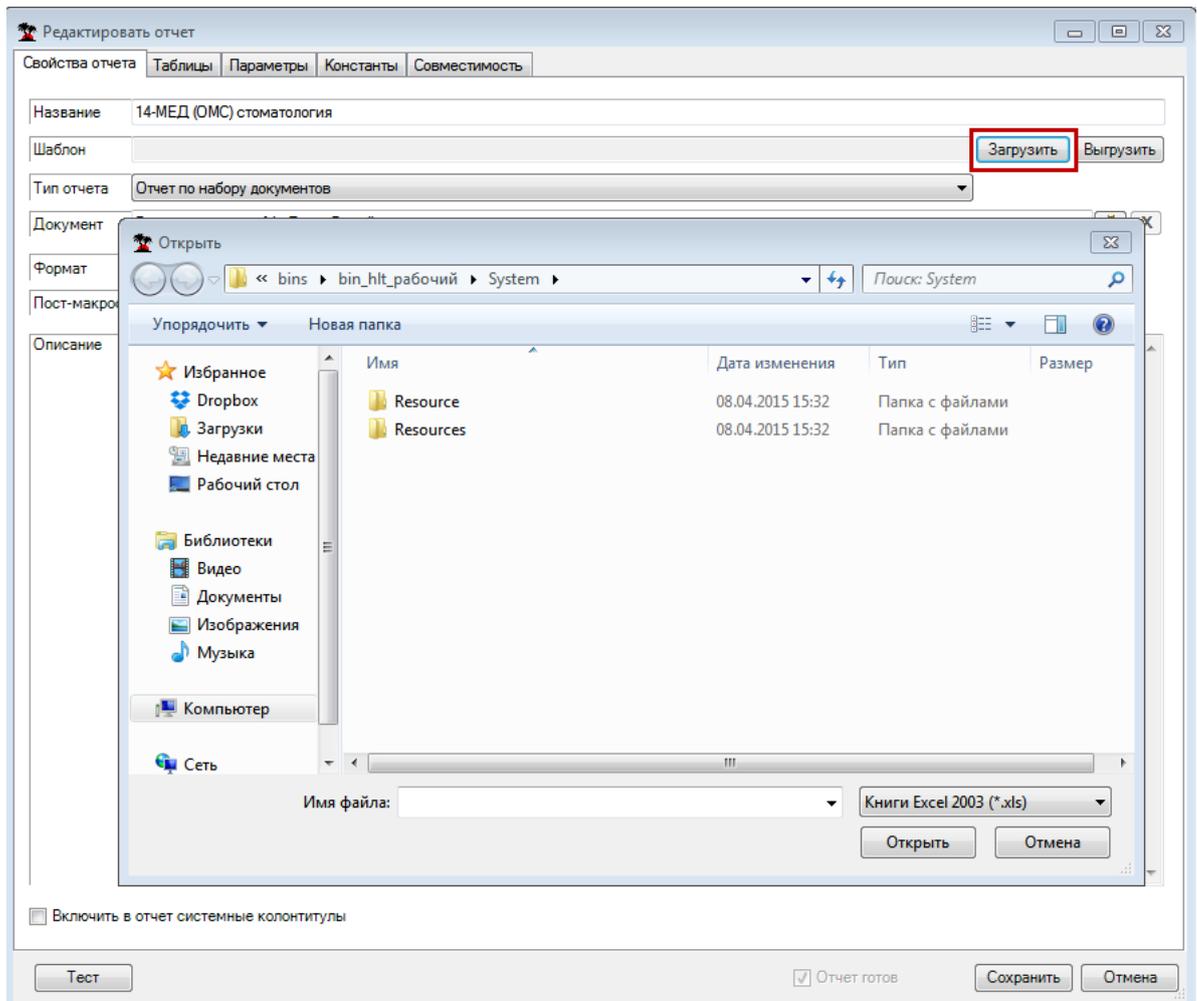


Рисунок 95. Загрузка шаблона отчета

А.1.2. Параметры отчетов

В момент формирования отчетов происходит подстановка значений параметров в текст SQL – запросов и в созданную по шаблону форму вместо синтаксических производных от имени параметра. Если непосредственно перед формированием отчета необходимо участие пользователя для инициализации значений параметров, то такие параметры считаются интерактивными и описываются на закладке «Параметры». Другие параметры, значения которых инициализируются на основе пользовательских настроек, описываются на закладке «Константы». Третий вид параметров вовсе не требует описания в форме редактирования. Их имена и механизм подстановки значений заранее predetermined.

А.1.2.1. Интерактивные параметры

Значения интерактивных параметров будут «спрашиваться» у пользователя непосредственно перед формированием отчета в диалоговом окне «Параметры», сформированного динамически на основе описания.

Для описания интерактивных параметров необходимо на закладке «Параметры» добавить имя параметра в список, определить текст вопроса (если нет – то сформируется стандартное приглашение) и тип, который определяет набор элементов значения параметра (Рисунок 96).

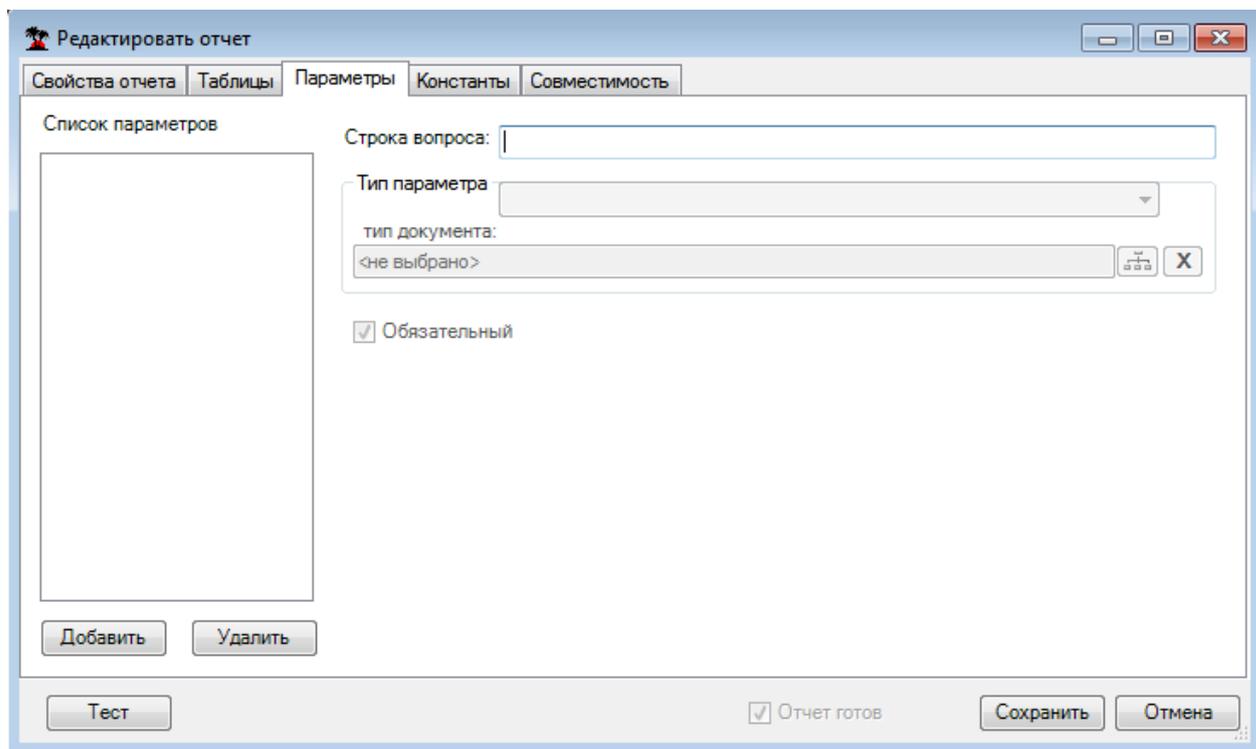


Рисунок 96. Редактирование параметров отчета

А.1.2.2. Доступ к значениям экземпляра документа и набора документов в отчетах

Доступ к значению текущего экземпляра документа осуществляется по правилам доступа к значению параметра с предопределенным именем «Document» и типом параметра Документ (тип документа соответствует текущему экземпляру).

Предопределенное для набора документов имя параметра «DocSet» используется только для доступа к значению WasInited (п. А.1.2.4).

А.1.2.3. Константы отчета – параметры на основе пользовательских настроек

Вид закладки для описания констант представлен на рисунке (Рисунок 97). В список констант нужно добавить имя параметра и заполнить для него атрибуты пользовательской настройки, из которой будет получено значение.

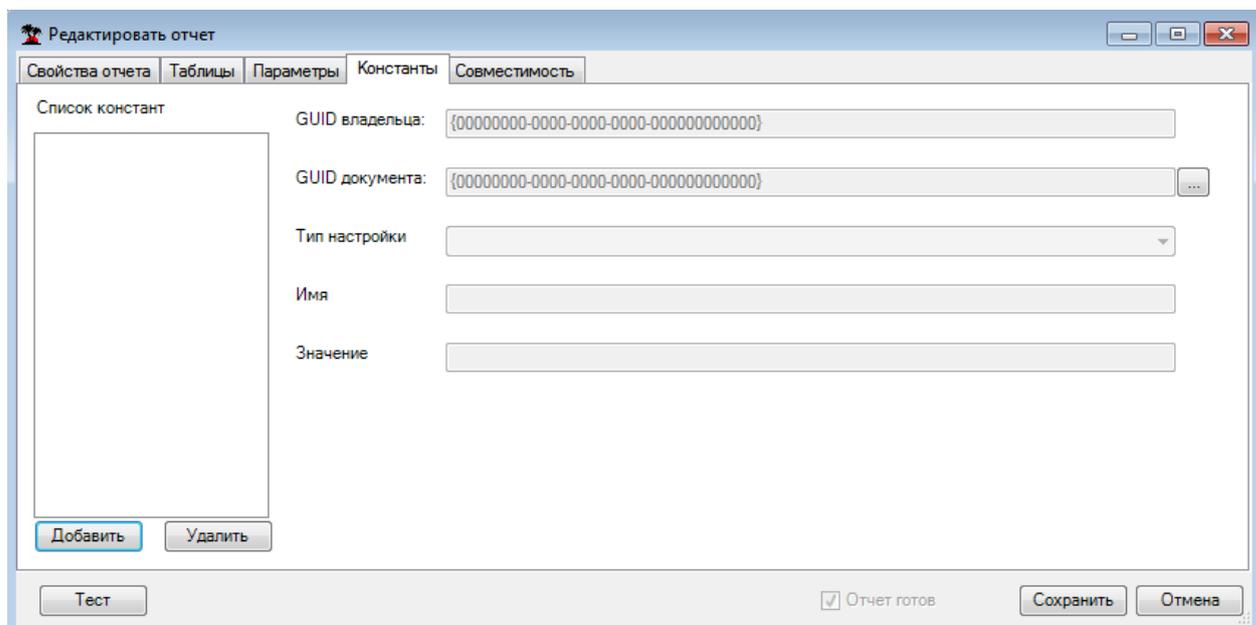


Рисунок 97. Описание констант

А.1.2.4. Доступ к значениям параметров. Особое значение WasInited

Доступ к значениям параметров осуществляется путем указания имени элемента после имени параметра:

<ИмяПараметра>.<ИмяЭлемента>

Состав элементов зависит от типа параметра. Для параметров простых типов (например, строка, дата) доступ к значению осуществляется через элемент .Value.

<ИмяПараметра>.Value

Для всех параметров, кроме специальных (см. ниже), проставляется значение особого элемента .WasInited (вольное сокращение от was initialized), которое равно «1» при успешной инициализации значения параметра и «0» – если инициализация не произведена.

<ИмяПараметра>. WasInited

Регистр букв в именах параметров и их элементов имеет значение!

Синтаксические конструкции доступа к значениям параметров применяются в тексте запроса с префиксом «@», а в шаблоне формы с префиксом «&&Param.» («@@Param.» – в колонтитулах).

А.1.2.5. Специальные параметры (зарезервированные имена)

В таблице (Таблица 4) перечислены специальные параметры, значения которых подставляются только в созданную по шаблону форму отчета и не подставляются в текст SQL-запроса. Формально тип этих параметров строковый, доступ к значению – .Value. Элемент «.WasInited» к специальным параметрам не применим.

Таблица 4. Специальные параметры

Имя параметра	Подставляемое значение
&&Params.Today	Текущая дата
&&Params.Time	Текущее время
&&Params.Now	Текущая дата-время
&&Params.UserFIO	Имя пользователя, запустившего отчет
&&Params.UserID	ID пользователя, запустившего отчет
&&Params.Connection	Имя подключения

А.1.3. Таблицы данных отчетов

Под термином «таблица» в системе отчетов понимается упорядоченный по строкам и столбцам набор значений (данных), получаемый в результате выполнения SQL-запроса и подставляемый в форму отчета согласно разметке в шаблоне.

Описание таблиц отчета подразумевает задание текста SQL-запроса с указанием подстановки параметров отчета, внесения разметки подстановки таблиц в шаблон формы отчета и описание группировки промежуточных итогов.

На закладке «Таблицы» (Рисунок 98) формы редактирования отчетов добавьте имя таблицы в список таблиц, заполните текст SQL-запроса и при необходимости сформируйте список групп (п. А.1.3.3) для промежуточных итогов.

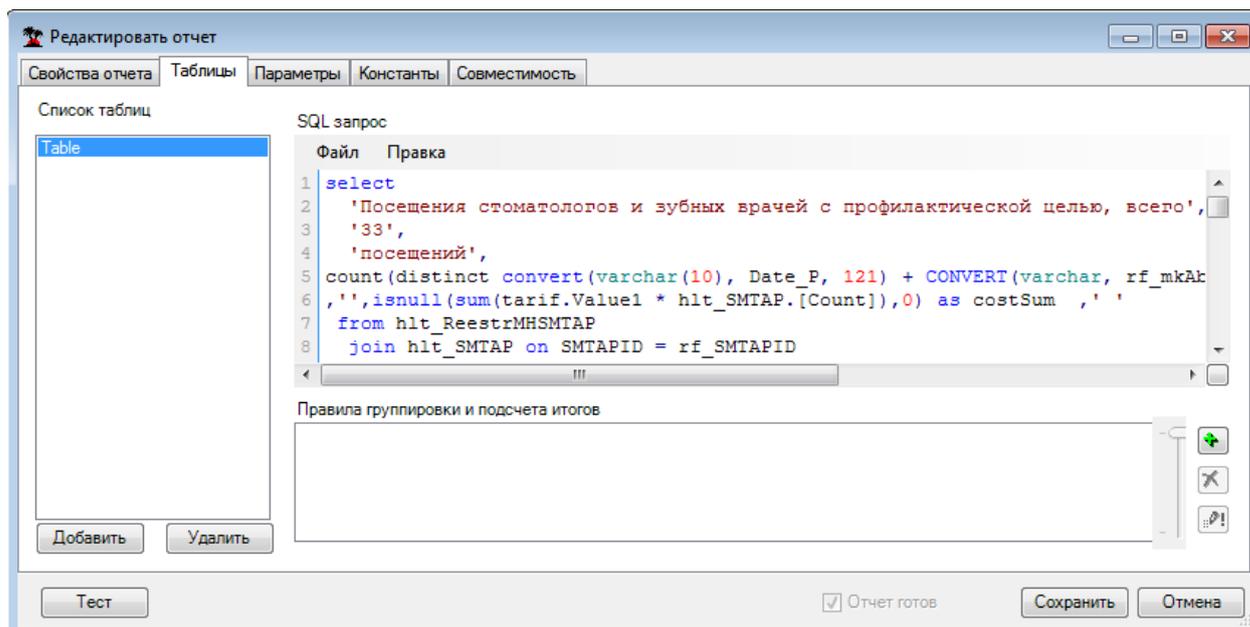


Рисунок 98. Закладка «Таблицы»

А.1.3.1. Запросы для получения таблиц данных

Рекомендации по написанию текста запроса несложные. Желательно задавать понятные (по-русски) имена колонок, используя ключевое слово «AS».

Перестраховка с использованием «convert(varchar...)» – устаревший прием. Попробуйте обойтись без него, поскольку теперь используется новая система преобразования типов SQL в типы данных электронной таблицы. Если используете группы

промежуточных итогов, то не забудьте задать необходимую сортировку ORDER BY. Соблюдайте синтаксис параметров отчета и собственно синтаксис SQL.

А.1.3.2. Правила задания подстановок значений параметров в текст запроса

Параметры отчета – это не параметры SQL! Замена вписанных в текст запроса синтаксических конструкций вида:

@<ИмяПараметра>.<ИмяЭлемента>

на соответствующие значения происходит еще до разборки синтаксиса SQL. Поэтому допустимо использование параметров в строковых константах, именах колонок и т.п., например:

convert(datetime, '@Period.EndDate', ...)

А.1.3.3. Группировка данных таблицы с подсчетом промежуточных итогов

Для описания группировки данных таблицы с подсчетом промежуточных итогов необходимо добавить (по соответствующей кнопке) уровень группировки в список «Группы» (Рисунок 99). Для каждого уровня группировки указывается поле группировки («Группировка по») и заполняется список агрегатных функций консолидации не группируемых полей («Промежуточные итоги по»). Необходимо заметить, что поля могут быть указаны как по именам, так и вводом порядкового номера поля (начиная с 0). При этом значение номера может превышать количество полей в запросе, если шаблон формы отчета предусматривает дополнительные (вычисляемые) поля таблицы.

Процесс группировки и подсчета промежуточных итогов осуществляется средствами электронной таблицы (Excel). Специальная разметка в шаблоне формы отчета для этого не требуется.

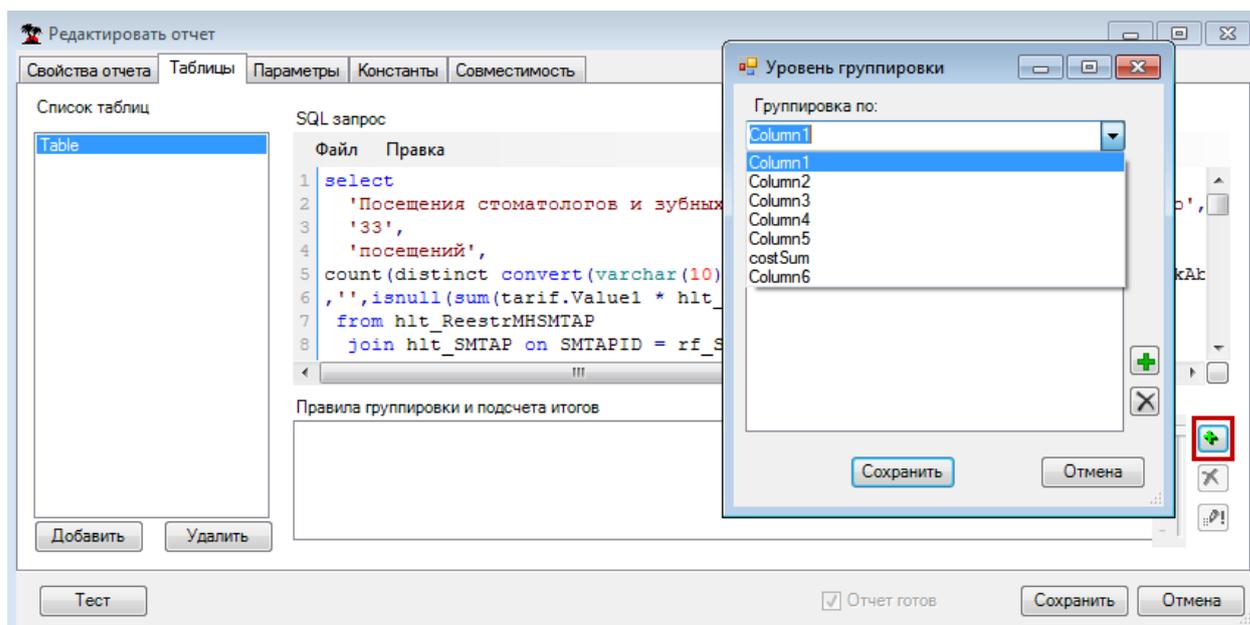


Рисунок 99. Добавление уровня группировки

А.1.4. Тестирование отчетов. Выгрузка/загрузка описателей отчетов.

Кнопка «Тест» на форме редактирования отчета позволяет попробовать сформировать отчет. Если тип отчета «по документу» или «по набору документов», то автоматически будет проинициализирован текущий документ (набор) произвольным экземпляром указанного типа документов. В остальном – так, как это будет в рабочем режиме (опрос интерактивных параметров и т.д.), но по окончании будет выдано окошко с сообщением о результатах тестирования.

Функции выгрузки в файл и загрузки из файла доступны во всплывающем контекстном меню на строке отчетов.

А.2. Правила создания Excel-шаблонов отчетов

А.2.1. Общие рекомендации

При создании шаблонов отчетов всегда следует стремиться к тому, чтобы вид документа на экране позволял удобно и быстро оценить его содержание, а вид «на бумаге» обеспечивал получение безупречной «твердой копии» без лишних телодвижений. Не забывайте пользоваться предварительным просмотром печатных страниц и уделять должное внимание их оформлению.

Убирайте ненужные листы из шаблона!

Давайте разнообразные осмысленные имена параметрам и таблицам отчета. Не допускайте синтаксических совпадений имен между собой и с зарезервированными именами («Table1» – «Table11», «Par» – «ParDate», «TimeOfVisit» – «Time»).

А.2.2. Подстановка значений параметров (&&Param.)

Если отчет имеет параметры, которые должен ввести пользователь, (интерактивные), значения этих параметров рекомендуется отобразить в заголовке документа над шапкой таблицы.

Значения параметров подставляются в отчет взамен внесённых в шаблон (как часть значения или формулы ячейки) синтаксических конструкций доступа к значениям параметра (п. А.1.2) с префиксом &&Param:

&&Param.<НазваниеПараметра>.<НазваниеЭлемента>

Если тип параметра – период, то конструкция принимает вид:

&&Param.<НазваниеПараметра>.BeginDate
(начало периода),

&&Param.<НазваниеПараметра>.EndDate
(конец периода)

если тип параметра – дата:

&&Param.<НазваниеПараметра>.Value

если тип параметра – документ:

&&Param.<НазваниеПараметра>.<ИмяПоля>

где <ИмяПоля> – название поля(элемента) документа–параметра.

Пример: Параметр TypeDelivery – документ Тип Поставки.

Для вывода названия выбранного типа поставки в шаблоне следует указать:

`&&Param.TypeDelivery.Name`

Для отчетов по документу существует параметр – тот документ (экземпляр), по данным которого был сформирован отчет.

При этом в шаблоне можно определить вывод любого поля этого документа, что очень удобно, например, при выводе шапки накладной (номер, дата документа, поставщик, покупатель и т.п.).

Для этого в значении нужной ячейки шаблона следует указать:

`&&Param.Document.<Имя поля>`

ВНИМАНИЕ! ВАЖНОЕ ЗАМЕЧАНИЕ: Во всех этих случаях регистр букв важен! Например, typedelivery и TypeDelivery – это разные параметры, а NUM и num – разные поля документа.

Подстановка констант и специальных параметров производится аналогично (см. п. А.1.2).

А.2.2.1. Подстановка значений параметров в колонтитулы (@@Param.)

К сожалению, синтаксис «&&Param» невозможно использовать для вставки значений параметров в колонтитулы. В этом случае используется похожая синтаксическая конструкция:

`@@Param.<НазваниеПараметра>.<НазваниеЭлемента>`

А.2.3. Рекомендации по оформлению шапки таблицы

Шапку таблицы предлагается выделять жирным шрифтом и жирными границами. Первая колонка обычно содержит порядковые номера записи (№ п/п). Дополнительный нижний ряд шапки содержит порядковые номера колонок (1, 2, 3, 4... – шрифтом на 1-2 меньшим основного шрифта таблицы) и не допускает объединения ячеек по вертикали. Автофильтр («Сервис» → «Данные» → «Фильтр» → «Автофильтр») следует поместить именно в этот ряд (Рисунок 100). Наличие такого нижнего ряда шапки особенно важно при использовании группировок с промежуточными итогами во избежание лишних диалогов. Часто имеет смысл повторять всю шапку или только нижний её ряд на каждом печатном листе («Файл» → «Параметры страницы» → «Лист» → «Сквозные строки»), а иногда и первую колонку (№ п/п) – в качестве сквозного столбца, если ширина отчета больше страницы.

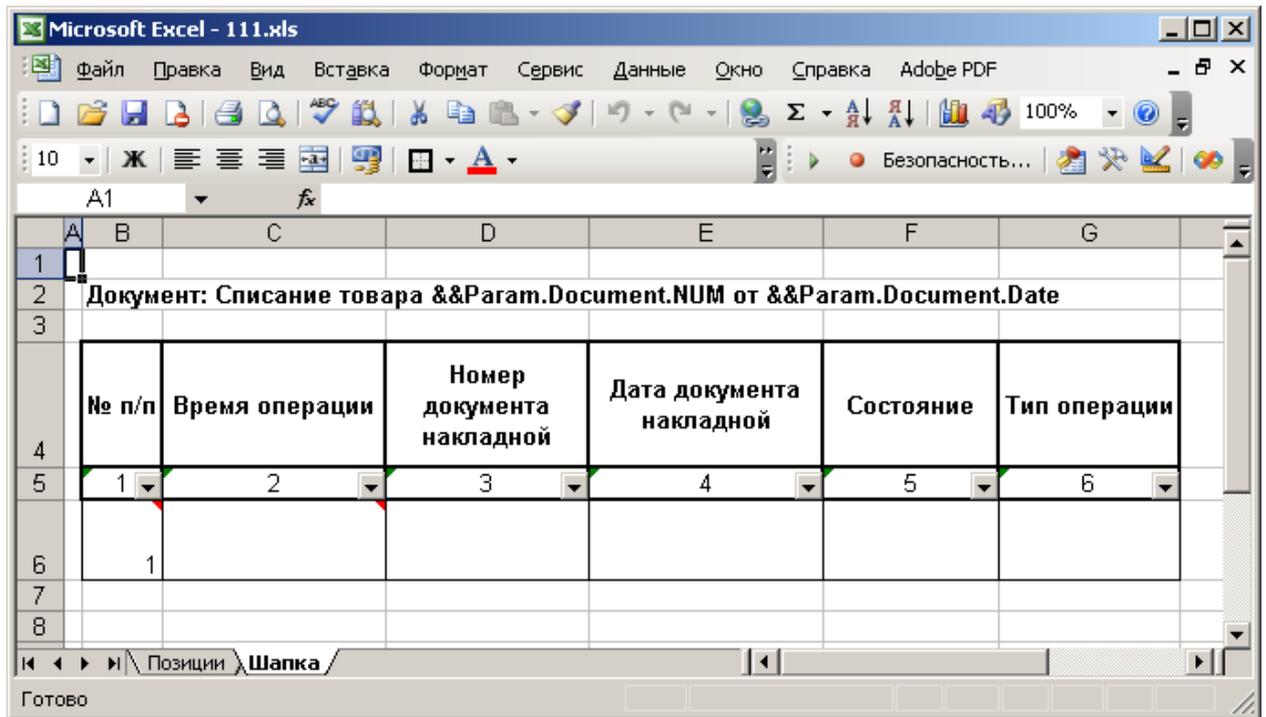


Рисунок 100. Шапка с автофильтром в нижнем ряду

При заполнении наименований колонок в шапке бывает предпочтительнее явно задать перевод строки (Alt+Enter), чем использовать автоматический перенос по словам.

Ширину колонок следует подобрать с учетом вероятных максимальных значений, чтобы не заставлять пользователя самостоятельно раздвигать колонки после того, как отчет уже напечатан на 576 листах. Автоподбор ширины колонки («Формат» → «Столбец» → «Автоподбор ширины») имеет смысл задавать только в случаях, когда не «мешаются» ячейки с переносом по словам, и когда автоматический подбор ширины не повлияет на размещение отчета на печатной странице. Дело в том, что при включенной опции «разместить не более чем 1 страницу в ширину» это может повлечь изменение масштаба, а при выключенной – риск вылезти за пределы страницы.

В начальном ряду таблицы (следующий ряд с примечаниями &&Tab. – см. ниже) следует осуществить автоподбор высоты строки («Формат» → «Строка» → «Автоподбор высоты»), кроме того в одной из ячеек этого ряда часто полезно закрепить области прокрутки («Окно» → «Закрепить области»).

Для наиболее правильного применения автофильтров в шаблонах XL-отчетов, чтобы они успешно работали и под OpenOffice, чтобы в списки предлагаемых критериев фильтрации попадали все необходимые значения, чтобы итоги выводились с учетом фильтрации, необходимо запомнить несколько вещей (Рисунок 101):

1. вставляйте между итогами и таблицей (маркировкой &&Tab.) пустую строку;
2. включайте ее вместе с итогами в область &&TabEnd;
3. отмечайте область фильтрации, как показано на рисунке, жёлтым цветом (Рисунок 101). Область должна совпадать по ширине с областью &&TabEnd, включать пустую строку, но не включать строку итогов;
4. в формулах итогов лучше использовать функцию «Промежуточные итоги» (Subtotals) с диапазоном от строки &&Tab до пустой строки.

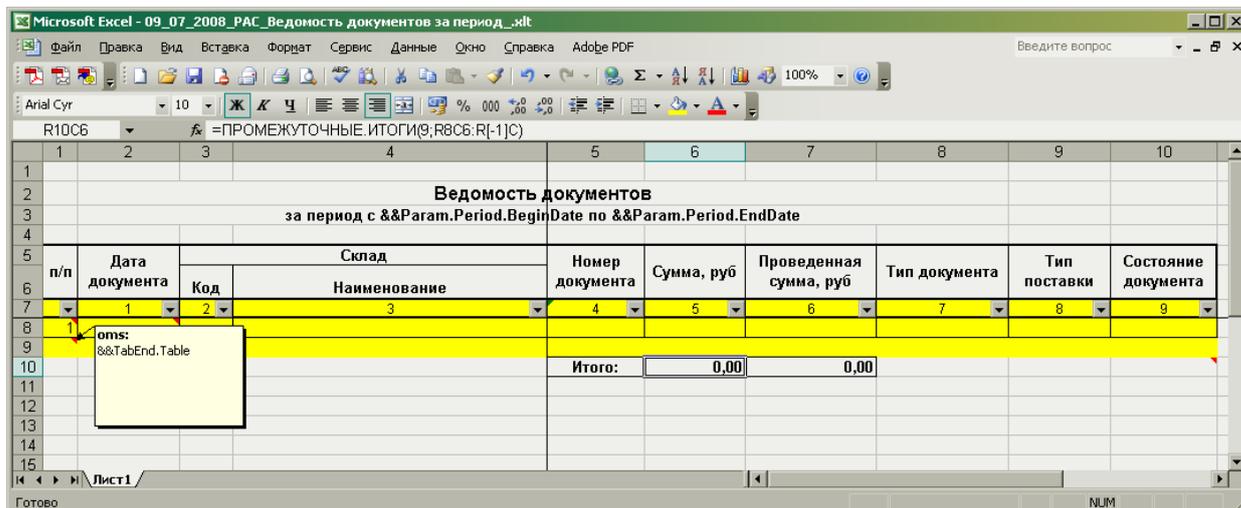


Рисунок 101. Пример разметки автофильтра в шаблоне отчета

A.2.4. Описатель начала подстановки данных таблицы отчета (&&Tab.)

Ячейка, помеченная в шаблоне примечанием, содержащим конструкцию вида:

&&Tab.<ИмяТаблицы>

является местом начала подстановки данных соответствующей таблицы отчета (см п. А.1.3).

Размер подстановки зависит от количества колонок и строк (записей) в наборе данных результата выполнения запроса.

A.2.5. Порядковые номера записей в таблице (.RecordNumber).

Ячейка, помеченная в шаблоне примечанием, содержащим конструкцию вида:

!--<ИмяТаблицы>.RecordNumber

расположенная в одном ряду с (обычно непосредственно перед) ячейкой начала таблицы (&&Tab.), является местом начала подстановки столбца с номерами строк (записей) соответствующей таблицы отчета. Значением такой ячейки должен быть начальный номер (1).

A.2.6. Диапазон подстановки полей (&&Details)

Альтернативой подстановки данных таблицы отчета целиком является подстановка значений указанных полей таблицы в помеченный диапазон ячеек шаблона.

Ячейки, помеченные в шаблоне примечанием, содержащим:

&&Details

а в значении содержащие:

!--<ИмяТаблицы>.<ИмяПоля>

являются местом подстановки соответствующих значений полей таблицы и образуют диапазон ячеек, повторяющийся в отчете с каждой записью таблицы.

При этом в диапазон могут входить ячейки, помеченные &&Details, но не содержащие !--<ИмяТаблицы>.<ИмяПоля>. Подстановка в них не производится, но они повторяются вместе с Details-диапазоном.

А.2.7. Для чего и как нужно применять &&TabEnd.

Помещенные в примечаниях ячеек шаблона конструкции вида

&&TabEnd.<ИмяТаблицы>

позволяют предотвратить перекрытия данными таблицы содержимого (значений, формул, разметки) лежащих ниже ячеек.

Не обязательно помечать &&TabEnd каждую ячейку. Прямоугольная TabEnd-область определяется границами с минимальной по максимальную строки и с минимального по максимальный столбец помеченных &&TabEnd ячеек. Значения и примечания в ячейках ниже TabEnd-области также не пострадают, но правильный пересчет относительных формул – только внутри TabEnd-области.

Конструкции &&TabEnd можно совмещать в одном примечании с описателями сводных таблиц (обычно с &&PivotTableEnd).

А.2.8. Описатели, используемые для задания сводных таблиц

А.2.8.1. Разметка региона – источника сводной таблицы (&&PivotTableBegin и &&PivotTableEnd)

Конструкции &&PivotTableBegin и &&PivotTableEnd в примечаниях ячеек шаблона отмечают соответственно начало (левый верхний угол – выше шапки) и конец (правый нижний угол) региона – источника сводной таблицы. Размещение этих конструкций в значениях ячеек является устаревшим, но ещё допустимым.

Напомним, что такие конструкции можно совмещать в одном примечании с &&TabEnd.

А.2.8.2. Строки, колонки и данные сводных таблиц

Конструкции

&&PivotTableRow[{. ФункцияSubtotals}]

&&PivotTableColumn[{. ФункцияSubtotals}] и

&&PivotTableData[. Функция]

в примечаниях ячеек шапки источника сводной таблицы определяют соответственно источники строк, столбцов и данных сводной таблицы.

Функция – название функции консолидации (Таблица 5). Для данных функций по умолчанию является сумма.

[{. ФункцияSubtotals}] – необязательное определение промежуточных итогов строк или столбцов сводной таблицы. Для строк и столбцов можно указать несколько функций для промежуточных итогов, по умолчанию промежуточные итоги не подсчитываются.

Таблица 5. Функции промежуточных итогов

--	--

Функция	Операция
Average	Среднее арифметическое
Count	Количество
CountNums	Количество чисел
Max	Максимум
Min	Минимум
Product	Произведение
StDev	Смещенное отклонение
StDevP	Несмещенное отклонение
Sum	Сумма
Var	Смещенная дисперсия
VarP	Несмещенная дисперсия

А.2.8.3. Приемник сводной таблицы

По умолчанию результат сводной таблицы размещается на новом листе. Однако, можно заранее подготовить в шаблоне отчета место приемника сводной таблицы, пометив начальную ячейку примечанием &&PivotTableDestination.

А.2.8.4. Построение диаграммы на основе сводной таблицы

Наличие примечания &&PivotTableMakeDiagram в одной из ячеек шаблона приведет к построению в отчете диаграммы на базе сводной таблицы.

А.2.9. Пост-макросы в Excel-отчетах

На закладке свойств формы редактирования отчетов (Рисунок 92) имеется возможность задать имя макроса, который должен выполняться после завершения формирования отчета (пост-макрос). Этот макрос должен быть размещен в модуле шаблона отчета в виде процедуры без параметров (параметры опускаются – missing) и может выполнять различные функции (например, дополнительное форматирование данных отчета).

При написании макроса разработчик шаблона отчета должен учитывать следующее.

В момент запуска макроса автоматический пересчет формул, отображение и интерактивность приложения электронной таблицы отключены. При необходимости (например, анализа значения – результата формулы) – включайте самостоятельно.

Для того чтобы макрос обрабатывал как в среде Excel, так и Open Office Calc (поскольку MS VBA и Basic Open Office совместимы в основе языка и не совместимы объектные модели Excel и Open Office Calc), необходимо организовать отдельные ветки исполнения макроса для Excel и Open Office Calc примерно следующим образом:

```
Public Function isXL() As Boolean
    Dim ret As Boolean
    ret = False
On Error GoTo eee
    Dim path As String
    path = Application.path
    ret = True
eee:
    isXL = ret
```

End Function

```
Public Sub MyPostMacro()  
    If isXL Then  
        'Код для Excel  
        MyPostMacro4XL  
    Else  
        'Код для Open Office Calc  
        MyPostMacro4OO  
    End If  
End Sub
```

```
Sub MyPostMacro4XL  
    '....  
End Sub
```

```
Sub MyPostMacro4OO  
    '....  
End Sub
```

Приложение Б Подсистема DTS

Б.1. Назначение подсистемы экспорта-импорта

Подсистема экспорта-импорта, она же подсистема DTS (Data Transformation Service – служба преобразования данных) предназначена для загрузки, выгрузки, анализа и преобразования данных, хранящихся в форматах XML, DBF, а также в СУБД MSSQL.

Средства подсистемы обеспечивают создание, редактирование и исполнение сценариев переноса и преобразования данных. В терминах системы сценарий преобразования данных носит название «Протокол».

Б.2. Протокол

Протокол – центральное понятие системы. Протокол – это сокращенный вариант полного названия «Протокол загрузки-выгрузки данных». Фактически, протокол является некоторой последовательностью действий по преобразованию данных. Также он включает в себя функции анализа данных и содержит ряд средств, с помощью которых осуществляется взаимодействие с пользователем, необходимых для управления процессом выполнения.

Иногда используют разделение на протоколы загрузки и выгрузки. Протоколы загрузки выполняют загрузку XML или DBF файлов в базу данных. Протоколы выгрузки выполняют выгрузку из базы данных в файлы. В то же время, возможно создать протокол, не являющийся ни протоколом загрузки, ни протоколом выгрузки – он может выполнять, к примеру, конвертацию данных в пределах БД или осуществлять некоторый анализ.

Б.2.1. Управление протоколами

Протоколы приложения хранятся в базе данных. Управление протоколами осуществляется из приложения администрирования на вкладке «Протоколы» (Рисунок 102). Здесь возможно создание, редактирование, удаление протоколов, а также их загрузка и выгрузка.

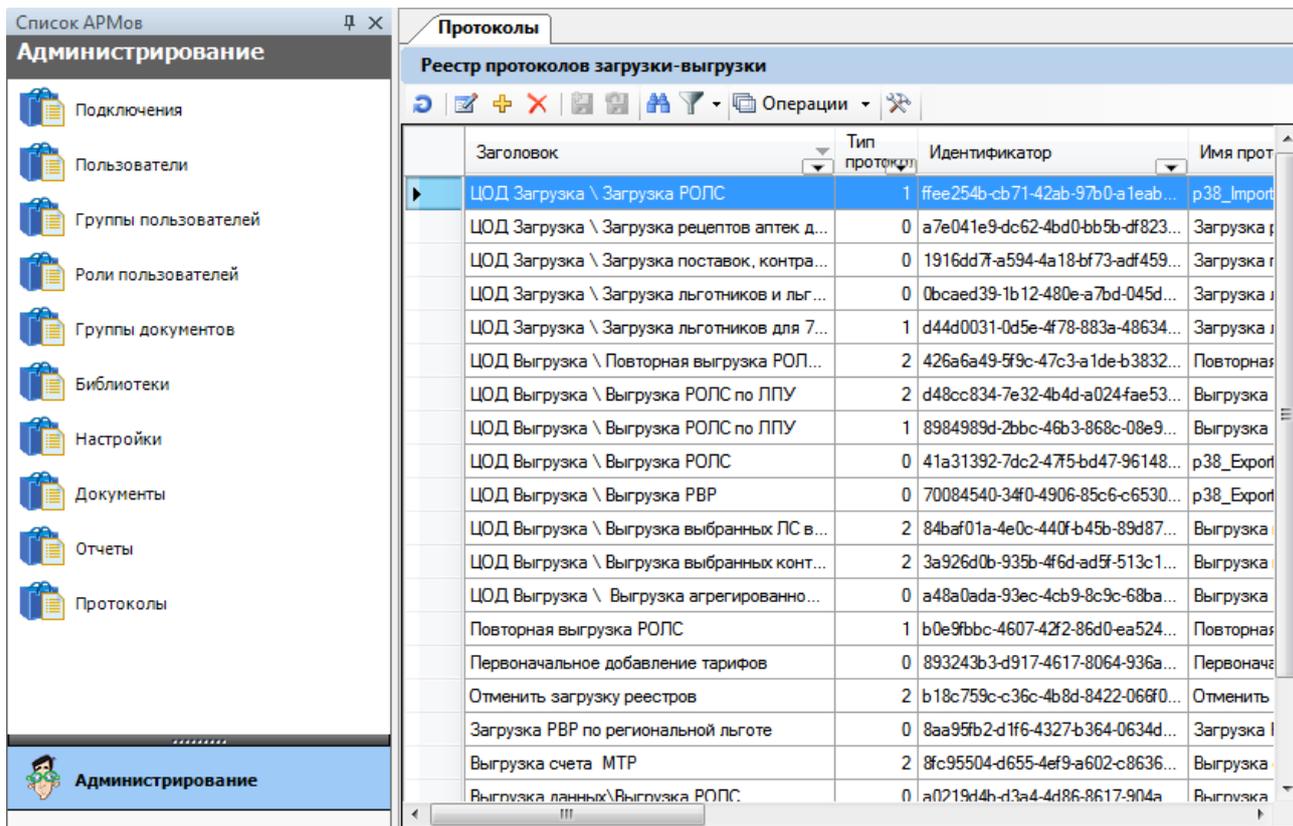


Рисунок 102. Вкладка «Протоколы» в приложении администрирования

Операции загрузки и выгрузки доступны из контекстного меню списка протоколов. Создание и редактирование протокола осуществляется в диалоговом окне (Рисунок 103).

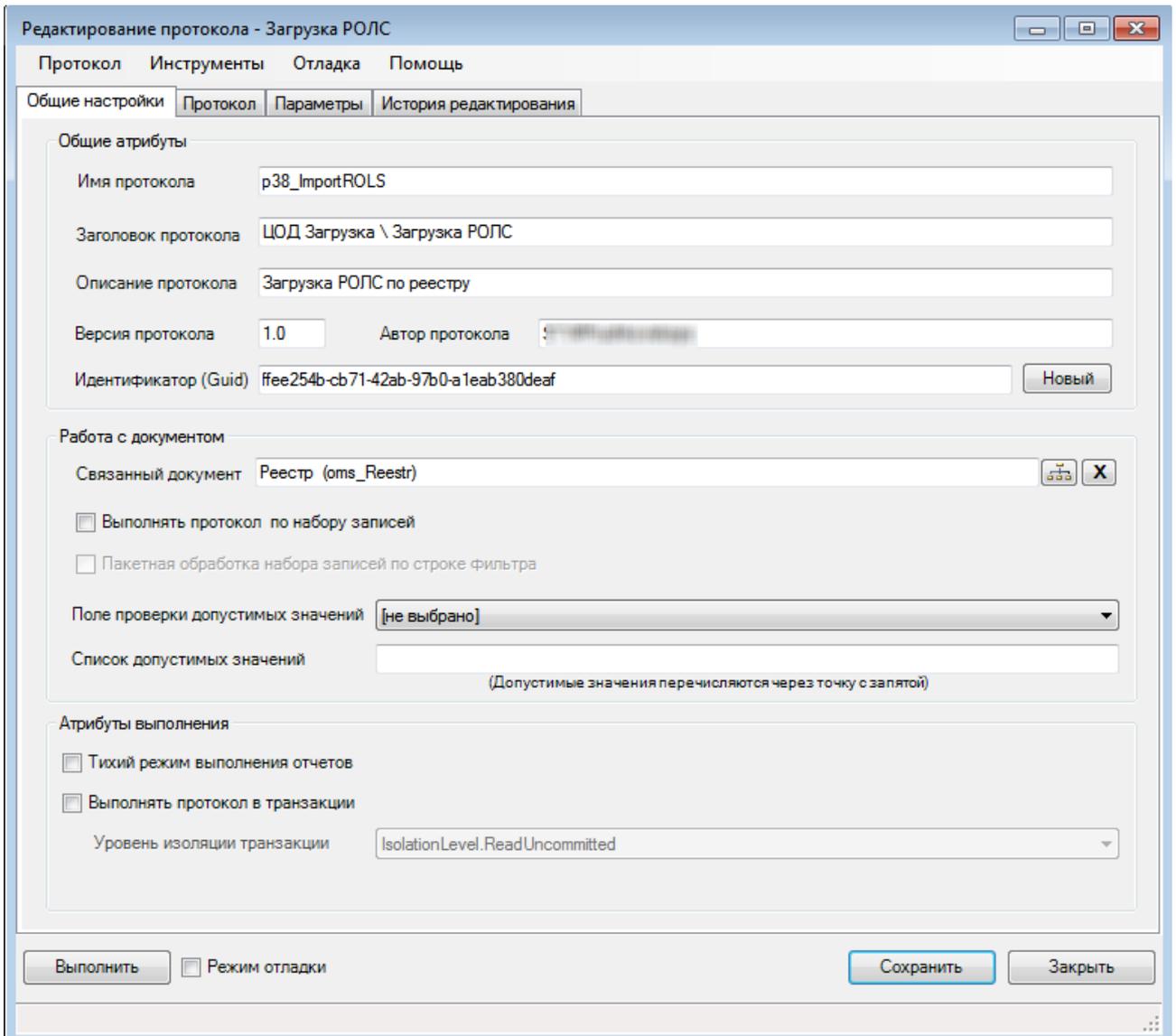


Рисунок 103. Диалог редактирования протокола

Б.2.2. Структура протокола

В протокол входит множество элементов, которые можно условно разделить на следующие группы:

1. Базовый набор таких атрибутов, как название, идентификатор и пр.
2. Свойства интеграции, обеспечивающие привязку протокола к документам приложения.
3. Свойства поведения протокола в процессе его исполнения, такие как «Тихий режим выполнения отчетов» и «Выполнить протокол в транзакции».
4. Набор сценариев преобразования, анализа и взаимодействия с пользователем, последовательность исполнения которых образует процесс выполнения протокола. Набор действий может включать в себя любое число следующих элементов:
 - Параметр – некоторая единица данных (строка, число, документ и т.п.), вычисляемая в процессе выполнения протокола или запрашиваемая у пользователя.
 - Скрипт – sql-запрос для выполнения дополнительных действий в базе данных, в ходе исполнения протокола.
 - Отчет – средство анализа данных, предоставляющее форматированный результат, а также, при необходимости, позволяющее прервать выполнение протокола.

- Загрузчик данных, он же источник данных или просто "Источник" – правило переноса данных из XML-файла, DBF-файла или постоянной таблицы базы данных в одну или несколько временных таблиц.
- Выгрузчик данных, он же приемник данных или просто "Приемник" – правило переноса данных из временных таблиц базы данных в постоянные таблицы, XML-файлы или DBF-файлы.

Б.2.3. Выполнение протокола

Протокол выполняется в контексте базы данных MSSQL. Для временного хранения «сырых» данных и трансформации используются временные таблицы. Упрощенно процесс выполнения протокола можно описать следующим образом:

У пользователя запрашиваются все данные, необходимые для выполнения. Это могут быть файлы для загрузки и какие-либо параметры. Затем один или несколько загрузчиков (из файлов или таблиц БД) формируют временные таблицы с данными. Далее, один или несколько выгрузчиков выполняют выгрузку данных из временных таблиц в постоянные таблицы БД или файлы. В ходе процесса, между описанными шагами данные могут быть подвергнуты дополнительным преобразованиям с помощью скриптов и анализу с помощью отчетов. Отчеты позволяют показать пользователю форматированные результаты анализа, а также предоставить пользователю выбор – продолжить или прервать выполнение протокола.

Полная последовательность выполнения протокола выглядит следующим образом:

- 1) В случае наличия XML или DBF загрузчика у пользователя запрашиваются файлы для загрузки. На этом же этапе xml-загрузчик может самостоятельно пытаться проанализировать перечень файлов и предлагать их для загрузки.
- 2) Запрашиваются невычисляемые (пользовательские) параметры.
- 3) Протокол инициализирует вычисляемые параметры.
- 4) Запускаются на выполнение отчеты, которые могут выполнить предварительный анализ.
- 5) Исполняются заданные скрипты.
- 6) Запускается обработка источников.
- 7) Исполняются заданные скрипты.
- 8) Отрабатываются отчеты для анализа результата работы источников.
- 9) Запускается обработка приемников.
- 10) Исполняются заданные скрипты.
- 11) Выполняются отчеты для финального анализа.
- 12) Происходит очистка временных таблиц. В журнале ДТС фиксируется факт выполнения протокола. Пользователю предоставляется полный лог выполнения, а также возможность перезапустить протокол.

Следует отметить, что некоторые шаги из приведенного описания могут не выполняться, если нет соответствующих им элементов (скриптов, отчетов, параметров и т.д.).

Б.3 Основные свойства протокола

Протокол содержит большое количество свойств, которые можно разделить на несколько групп для удобства их описания. Данный раздел содержит полный перечень свойств и атрибутов протокола.

Б.3.1. Общие свойства

Просмотр и редактирование общих свойств протокола выполняется на вкладке «Общие настройки» (Рисунок 104).

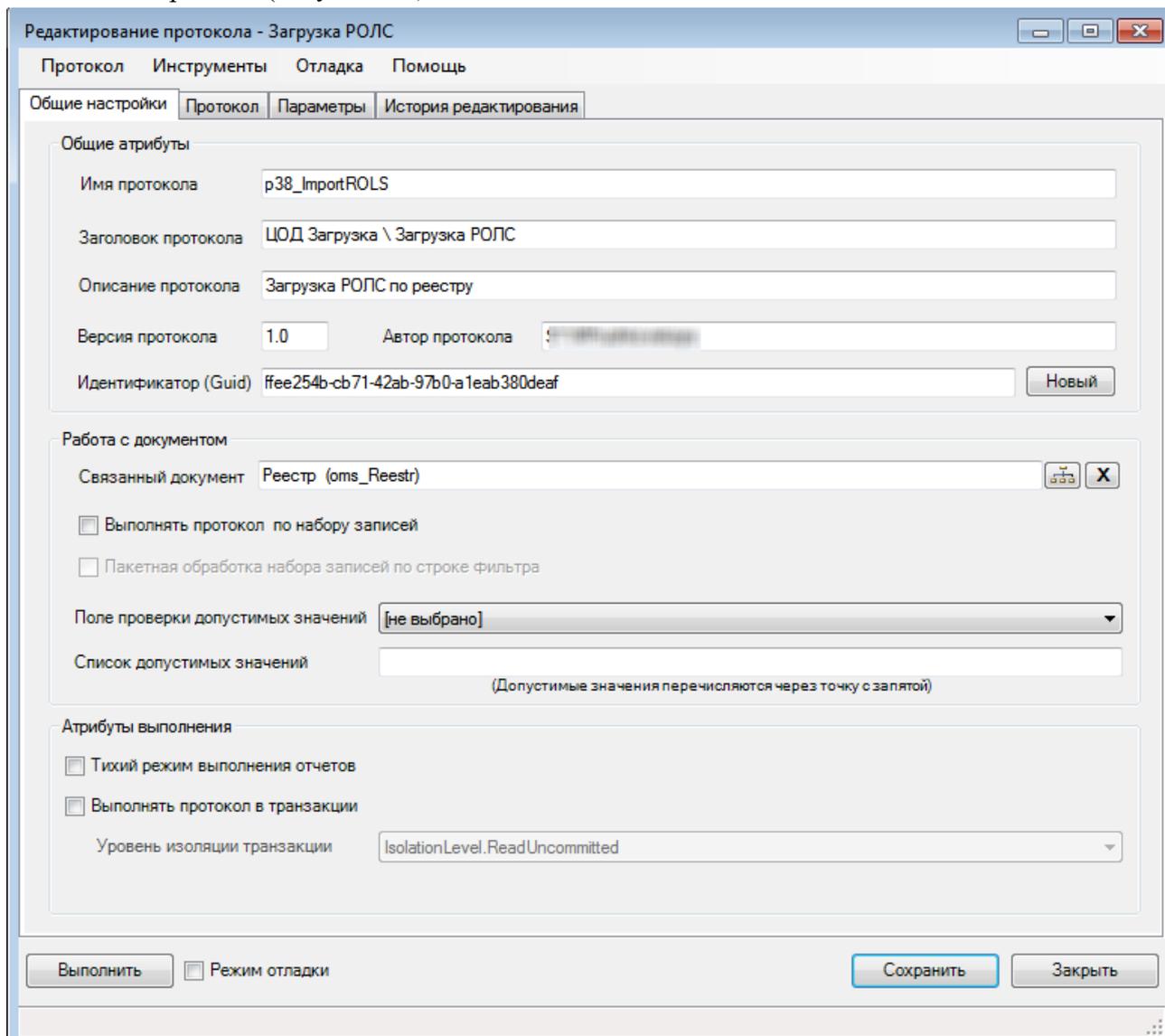


Рисунок 104. Диалог редактирования протокола

«Имя протокола» – имя, используется для идентификации одного и того же протокола в различных приложениях. Не отображается в интерфейсе приложения.

«Заголовок протокола» – имя протокола в интерфейсе данного приложения, включая путь к протоколу в меню. В качестве разделителя используется символ «\». К примеру, заголовок «Мониторинг\Выгрузка карт здоровья» означает, что протокол вызывается из подпункта меню «Мониторинг» и называется «Выгрузка карт здоровья».

«Описание протокола» – расширенное описание протокола, используется в качестве подсказки к пункту протокола в меню.

«Идентификатор» – идентификатор протокола уникальный в пределах базы данных. Внимание! Проверка на существование протокола в базе данных выполняется именно по идентификатору. Как следствие, изменение идентификатора с последующим сохранением протокола приведет к появлению копии протокола. Загрузка протокола при совпадении идентификаторов предложит переписать протокол или сменить идентификатор (Рисунок 105).

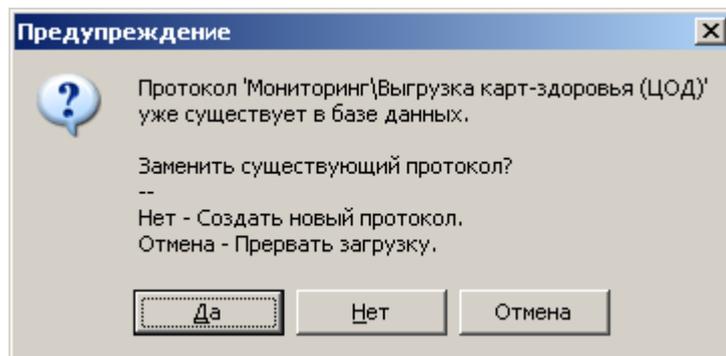


Рисунок 105. Предупреждение о совпадении идентификаторов при загрузке протокола

«Версия протокола» – специальное поле, позволяющее указать версию протокола. Версия позволяет принимать решение в случае существования нескольких модификаций одного и того же протокола.

Б.3.2. Привязка к документу

По типу привязки к документам протоколы можно разделить на три типа:

- Общий протокол – не привязан к конкретному документу, в интерфейсе приложения доступен из главного меню.
- Протокол по документу – привязан к документу, выполняется по одной записи (документу), в интерфейсе приложения доступен из контекстного меню.
- Протокол по набору документов – привязан к документу, может быть запущен как по одной, так и по нескольким записям (документам). В интерфейсе протокол доступен как из контекстного меню, так и из панели инструментов.

«Связанный документ» – поле для привязки протокола к документу. У общего протокола связанный документ будет не указан.

«Выполнять протокол по набору записей» – признак того, что протокол можно запускать по набору документов. Если признак установлен – протокол выполняется по набору документу, не установлен – по одному документу.

«Пакетная обработка набора записей по строке фильтра» – свойство, определяющее особый способ передачи набора записей в протокол. Когда свойство установлено, то при запуске протокола инициализируется служебный параметр «@WhereString», содержащий условие для выборки набора входных записей. Подробное описание порядка работы с параметрами смотрите в п. Б.4.1.

Свойство «Поле проверки допустимых значений», совместно со свойством «Список допустимых значений» позволяют создать правило, проверяющее разрешен ли запуск протокола по выбранному документу или нет. Для создания данного правила необходимо из выпадающего списка «Поле проверки допустимых значений» выбрать столбец, значения из которого будут использованы в проверке. В поле «Список допустимых значений» необходимо указать допустимые значения, используя точку с запятой в качестве разделителя.

В интерфейсе приложения пункт вызова соответствующего протокола будет доступен или недоступен в зависимости от результата проверки. В случае если такая проверка не требуется, не заполняйте данные поля.

Б.3.3. Поведенческие свойства

«Тихий режим выполнения отчетов» – если данный флаг не установлен, при выполнении каждого отчета процесс выполнения будет приостанавливаться и ожидать действий пользователя. Если же данный флаг установлен, поведение отчетов меняется: Если по результату выполнения отчета пользователю будет доступен только один вариант действий (отменить или продолжить выполнение), данное действие будет применено автоматически, а сам отчет будет показан в составе общего лога выполнения. В противном случае потребуется реакция пользователя. Более подробно о ходе выполнения отчетов смотрите в разделе Б.4.3.

«Выполнять протокол в транзакции» – устаревший флаг, оставлен в целях совместимости, не рекомендуется включать. В случае, когда данное свойство включено, фактически в нескольких транзакциях выполняются только группы скриптов и отчетов. Источники и приемники выполняются вне транзакции. «Уровень изоляции транзакции» задается в одноименном поле.

Б.3.4. Режим отладки

В ходе своего выполнения протокол создает, а затем удаляет набор временных таблиц и представлений. Однако, если в ходе выполнения будут создаваться таблицы с заданными именами, то существует очень высокая вероятность возникновения конфликта имен в ходе параллельного выполнения протоколов на одной базе данных. По этой причине в процессе работы имена временных таблиц заменяются на произвольные, сгенерированные автоматически (формата tmpdts*). Из-за того, что подобное переименование, а также очистка временных таблиц крайне затрудняют отладку, был введен специальный режим отладки.

Режим отладки включается и выключается флажком «Режим отладки» в левом нижнем углу диалогового окна, рядом с кнопкой «Выполнить» (Рисунок 106). Протокол можно сохранять с флажком отладки, однако в этом случае будет показано специальное предупреждение.

Протокол в режиме отладки использует оригинальные (заданные в протоколе) имена временных таблиц, а также не удаляет их после завершения работы. Это позволяет отлаживать поведение протокола в процессе его разработки. Подобное поведение имеет побочный эффект. Повторный запуск протокола приведет к ошибке из-за того, что в базе данных уже будут временные таблицы с теми же именами. Временные таблицы протокола можно удалить из пункта меню «Отладка» → «Удалить отладочные таблицы протокола».

Также меню «Отладка» содержит пункт «Удалить все временные таблицы». Данный пункт удаляет все временные таблицы с именем формата tmpdts*, которые ранее могли остаться в базе данных по каким-то техническим причинам.

Б.3.5. История редактирования

Любой протокол содержит историю своего редактирования. Данный механизм, также как и версионность, позволяет отслеживать изменения протоколов. Историю редактирования протокола можно посмотреть на одноименной вкладке (Рисунок 106).

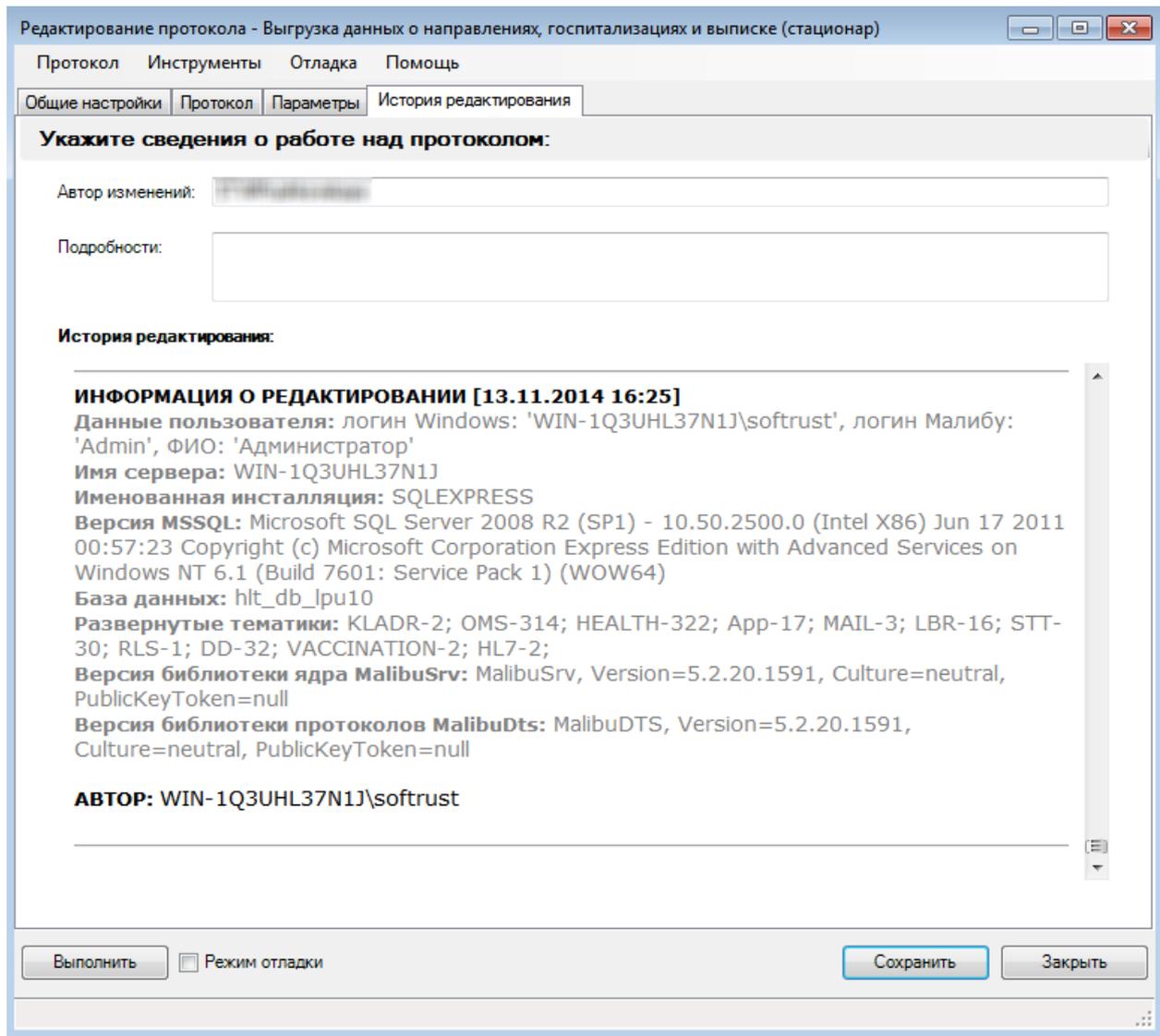


Рисунок 106. История редактирования протокола

Информация о редактировании записывается при сохранении протокола и включает в себя пакет технических сведений, а также необязательный комментарий пользователя. Если протокол был сохранен хоть один раз (за все время, пока у пользователя открыт диалог редактирования), это будет зафиксировано в истории редактирования.

При первом сохранении (за все время, когда открыт диалог редактирования) запись в историю протокола будет выполнена в любом случае. В дальнейшем, при сохранении протокола, запись в его историю редактирования будет выполнена, только если пользователь укажет комментарий.

Очистить историю редактирования можно с помощью пункта меню «Протокол» → «Очистить историю редактирования (в протоколе)». Однако после очистки история не будет пустой, т.к. факт очистки будет зафиксирован отдельной записью в истории редактирования.

Б.4. Исполняемые элементы протокола

Все исполняемые элементы протокола, за исключением параметров, находятся на вкладке «Протокол» (Рисунок 107). Управление параметрами вынесено на отдельную одноименную вкладку, подробнее смотрите раздел Б.4.1.

На вкладке «Протокол» находится список всех тех элементов (отчетов, скриптов, источников и приемников), последовательность исполнения которых составляет процесс

выполнения протокола. Последовательность, в которой показаны элементы, является строго упорядоченной и полностью отражает тот порядок, в котором они будут выполняться.

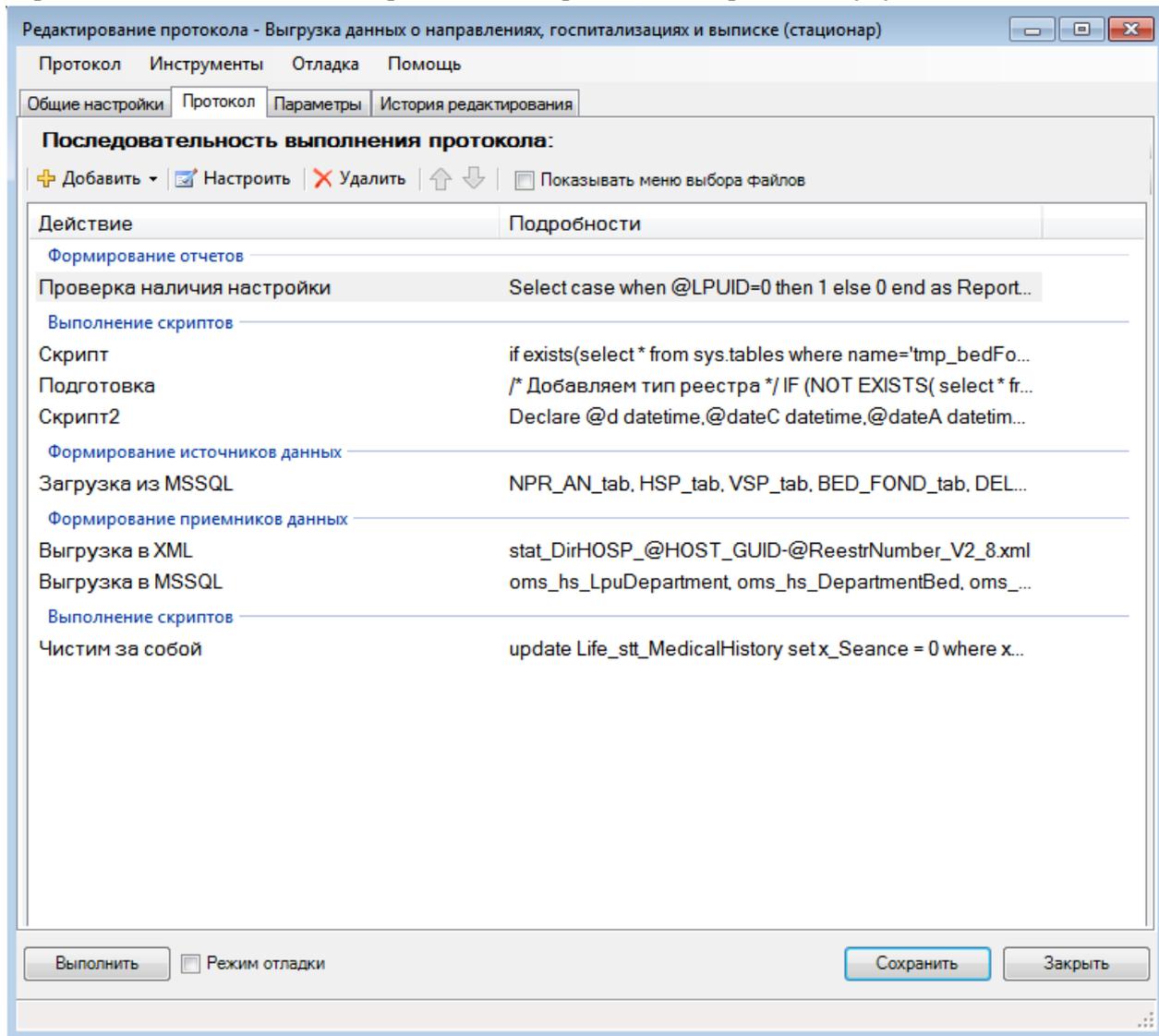


Рисунок 107. Последовательность выполнения протокола

На этой вкладке исполняемые элементы протокола можно создавать, удалять, редактировать, а также менять порядок их выполнения (Рисунок 108). Обратите внимание – порядок элементов можно изменить лишь в рамках той группы, к которой они относятся. Т.е. можно изменить взаимный порядок выполнения источников, но в то же время нельзя, к примеру, установить порядок выполнения приемников до источников – это действие не имеет смысла.

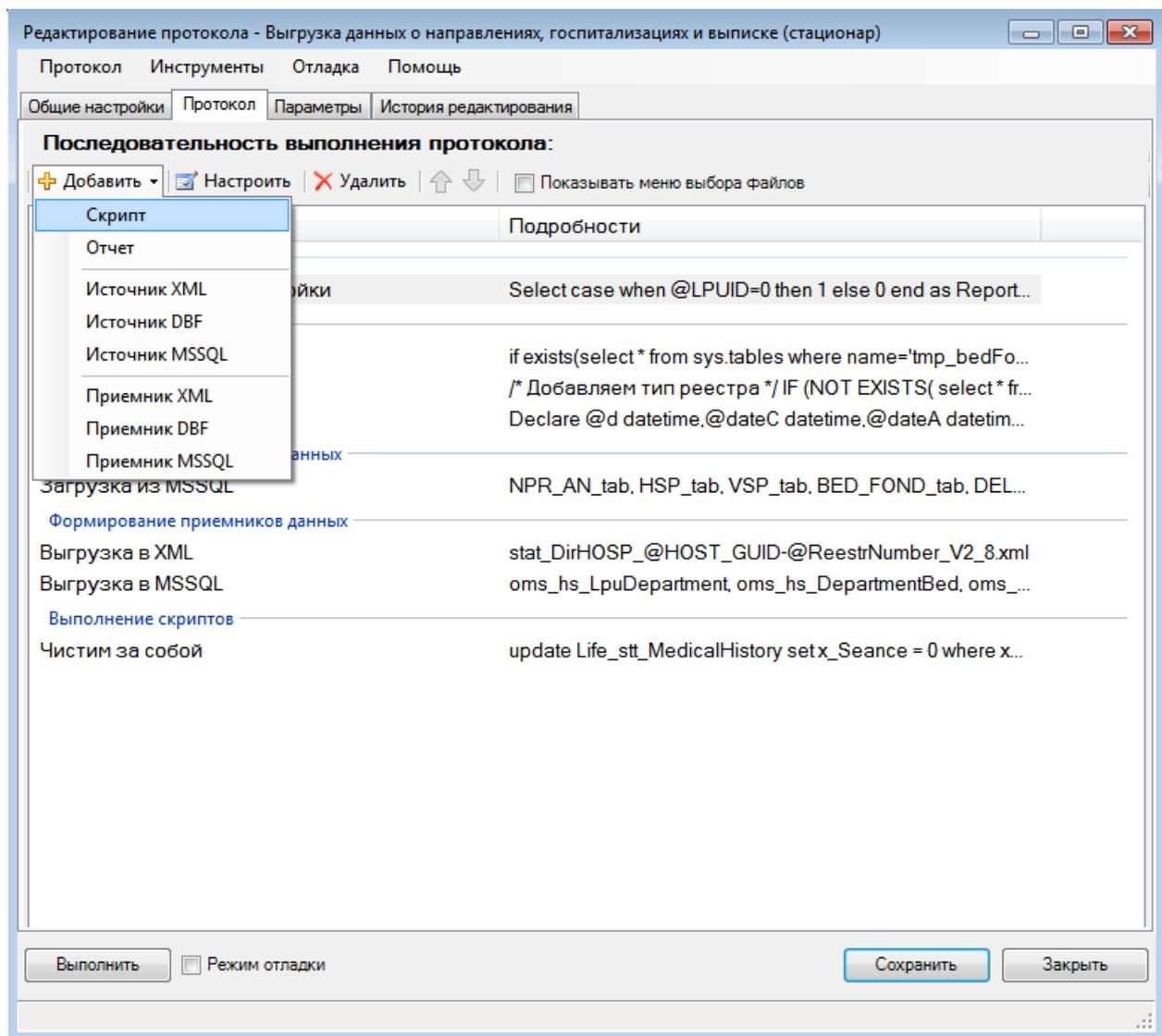


Рисунок 108. Добавление элемента в последовательность выполнения протокола

Б.4.1. Параметры

Параметр – это именованная единица данных (строка, число, документ и т.д.), вычисляемая в процессе выполнения протокола или запрашиваемая у пользователя. Параметры служат для того, чтобы задавать некоторые управляющие значения при запуске протокола и использовать их в процессе выполнения. Параметры протокола можно разделить на 3 вида:

- Служебные параметры – создаются и инициализируются протоколом автоматически в процессе выполнения. В редакторе параметров не отображаются.
- Задаваемые пользователем – объявляются в протоколе, значения запрашиваются у пользователя в начале выполнения протокола.
- Вычисляемые параметры – объявляются в протоколе, значения вычисляются автоматически, в момент после запроса параметров пользователя.

Б.4.1.1. Редактор параметров

Список параметров, а также управление ими, выполняется с помощью редактора на вкладке «Параметры» (Рисунок 109).

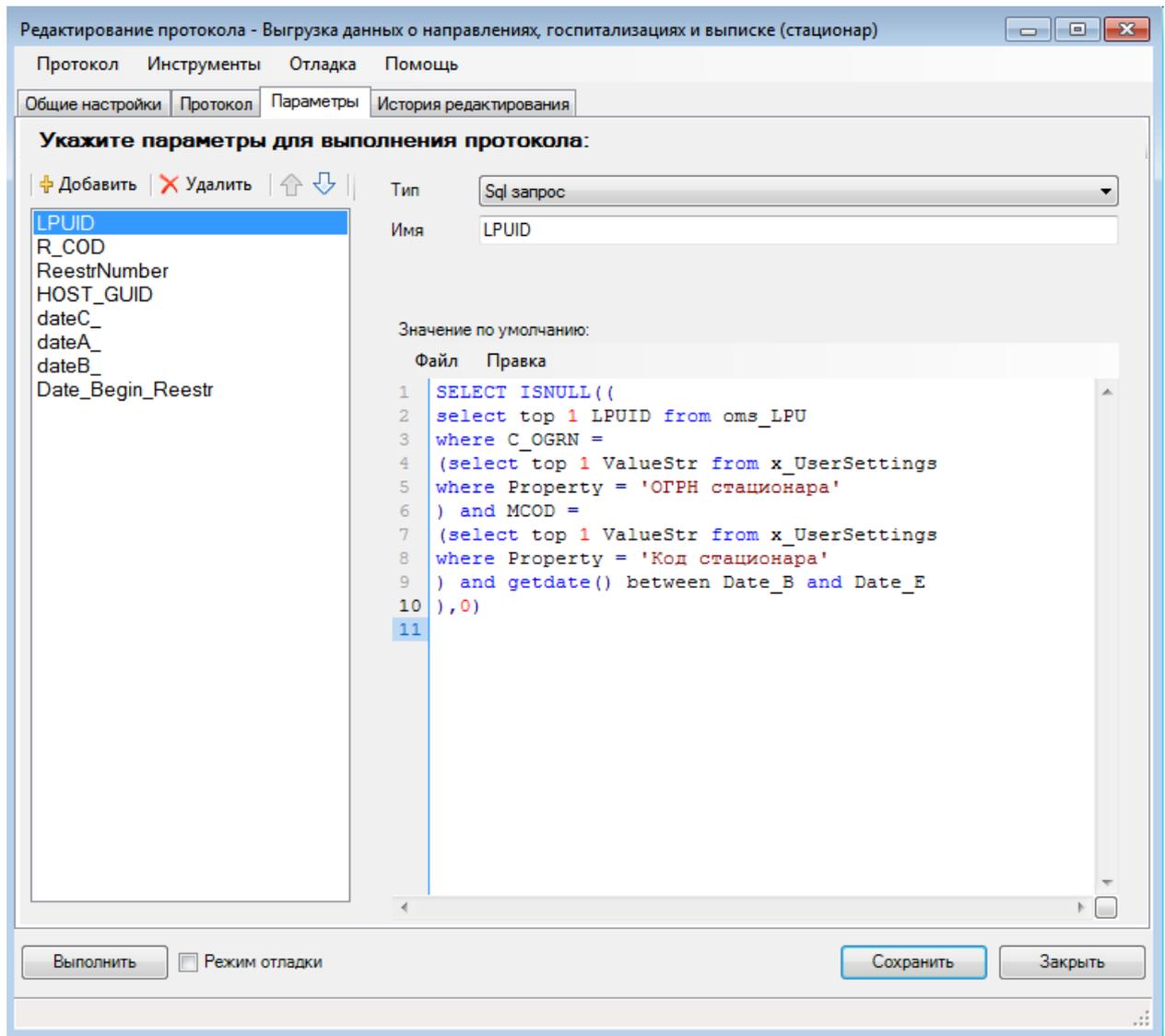


Рисунок 109. Управление параметрами протокола

Редактор позволяет создавать, изменять, удалять параметры, а также менять их порядок в списке, что соответственно сказывается на порядке их инициализации.

Каждому параметру указывается уникальное имя, с помощью которого к нему обращаются в протоколе. Для параметров, запрашиваемых у пользователя, также задаются заголовки, которые подскажут, какие данные и для чего необходимо ввести. Прочие свойства меняются в зависимости от типа параметра. Всего в протоколе доступно 5 типов параметров (Таблица 6).

Таблица 6. Типы параметров протокола

Тип параметра в редакторе	Поведения при выполнении	Значение	Значение по умолчанию
Строка	Запрашивается у пользователя	Простая строка, вводится пользователем	Поддерживается. Подставляется та же строка, что указана в поле «Значение по умолчанию»
Дата	Запрашивается у пользователя	Дата, указывается пользователем	Поддерживается. Дата указывается в поле «Значение по умолчанию»

			умолчанию»
Период	Запрашивается у пользователя	Составной параметр, указывается пользователем. Включает дату начала и дату конца периода: @PeriodParam.BeginDate, @PeriodParam.EndDate	Поддерживается. В поле «Значение по умолчанию» необходимо ввести 2 даты со знаком ";" в качестве разделителя
Документ	Запрашивается у пользователя	Составной параметр переменной структуры. Набор полей параметра повторяет поля документа.	Не поддерживается
Sql-запрос	Вычисляемый параметр, значение будет получено без участия пользователя	В поле «Значение по умолчанию» указывается Sql-запрос возвращающий 1 столбец и 1 строку.	Не имеет смысла

Б.4.1.2. Использование параметров

Параметры вне зависимости от их типа или вида могут быть использованы во всех sql-запросах протокола (источниках, приемниках, скриптах, отчетах), а также в шаблонах имен файлов. Для этого необходимо использовать имена параметров, которые в процессе выполнения протокола будут заменены соответствующими им значениями.

Для обращения к параметрам используется следующий синтаксис: @ИмяПараметра – если параметр имеет одно значение. Например, к параметру типа строка с именем MyParam обращаются как к @MyParam. Или же @ИмяПараметра.ИмяЗначения – если параметр имеет несколько значений. Например, для параметра типа период с именем PeriodParam обращаются как к @PeriodParam.BeginDate, @PeriodParam.EndDate – это использование соответственно даты начала и даты конца периода.

Внимание! Регистр в именах параметров имеет значение. Также обратите внимание, при создании параметра в редакторе символ @ в начале имени писать не требуется. Он является служебным маркером при замене параметров их значениями. В то же время, это не означает, что символ @ нельзя использовать в запросах иначе как с параметром. Требуется лишь внимательно относиться к именам переменных в Sql-запросах (они также объявляются с помощью символа @), чтобы те не совпадали с именами параметров.

Ниже представлены несколько примеров использования параметров различного типа.

Для параметров типа **Sql-запрос** в поле "Значение по умолчанию" вписывается sql-запрос, возвращающий 1 столбец и 1 строку. К примеру, запрос «select NEWID()».

Пример вызова параметра:

```
insert into table1(column1) values('{ @MySqlExp}')
```

Что выполнится на самом деле:

```
insert into table1(column1) values('{27467212-7C08-4C29-A172-2FDF9869E717}')
```

Для параметра типа «Период» используется следующий синтаксис вызова: @ИмяПараметра.BeginDate или @ИмяПараметра.EndDate. В запросе при выполнении будут подставлены соответственно значения начала и конца периода.

Пример вызова параметра:

```
select * from table1 where date >= Convert(datetime, '@MyPeriod1.BeginDate') and  
date <= Convert(datetime, '@MyPeriod1.EndDate')
```

Что выполнится на самом деле: (период был задан от 2006.02.01 до 2006.03.01)

```
select * from table1 where date >= Convert(datetime, '2006-02-01') and  
date <= Convert(datetime, '2006-03-01')
```

Для использования параметра типа «Дата» необходимо указать только его имя: @ИмяПараметра. В запросе при выполнении будет подставлена дата в строковом формате.

Пример вызова параметра:

```
select '@MyParam' as StartDate
```

Что выполнится на самом деле (была установлена дата 1900.01.01):

```
select '1900-01-01' as StartDate
```

Как было уже сказано ранее, параметры типа «Документ» являются составными. Набор значений данного параметра полностью соответствует набору полей документа, т.е. синтаксис вызова имеет вид: @ИмяПараметра.ИмяПоляДокумента.

Пример вызова параметра (DOCTOR) по документу «Врач», содержащего поле ФИО (FIO):

```
select '@DOCTOR.FIO' as DoctorFIO
```

Что выполнится на самом деле (выбран врач «Иван Федорович Крузенштерн»):

```
select 'Иван Федорович Крузенштерн' as DoctorFIO
```

Б.4.2.3. Служебные параметры

Все служебные параметры имеют predetermined имена. Редактор параметров не позволит создать параметр с тем же именем, что и служебный. В зависимости от типа протокола количество служебных параметров в нем может меняться. К примеру, если протокол запускается по документу, появляется специальный параметр @Document. Полный перечень всех служебных параметров смотрите в Таблица 7.

Таблица 7. Служебные параметры

Параметр	Условие объявления	Значение
@UserID	Объявлен всегда	Параметр типа строка. Содержит идентификатор пользователя, запустившего протокол.
@HostID	Объявлен всегда	Параметр типа строка, содержит идентификатор хоста. При отсутствии документооборота принимает значение -1
@File.FileName	Объявлен, если протокол выполняет загрузку файла.	Параметр типа строка. Имя файла, загрузка которого выполняется.
@File.FileExt	Объявлен, если протокол выполняет загрузку файла.	Параметр типа строка. Расширение файла, загрузка которого выполняется.
@File.FilePath	Объявлен, если протокол	Параметр типа строка. Путь к файлу,

	выполняет загрузку файла.	загрузка которого выполняется.
@WhereString	Объявлен, для протокола по набору записей с установленным флагом « Пакетная обработка набора записей по строке фильтра »	Параметр типа строка. Содержит условие фильтра, позволяющее выполнить выборку записей, по которым был запущен протокол.
@Document	Объявлен для протокола по документу.	Параметр типа документ. Содержит документ, по которому запущен протокол. Обращение к значениям выполняется так же как для параметра типа документ – @Document.ИмяПоляДокумента

Б.4.2. Скрипты

Скрипт – это sql-запрос для выполнения дополнительных действий в базе данных в ходе исполнения протокола. Его можно использовать, например, для сохранения в базе данных факта загрузки (или выгрузки) документов.

Сущность «Скрипт» состоит всего из трех элементов – необязательного имени, которое используется только в редакторе протокола, непосредственно sql-запроса и свойства, определяющего момент выполнения скрипта (Рисунок 110).

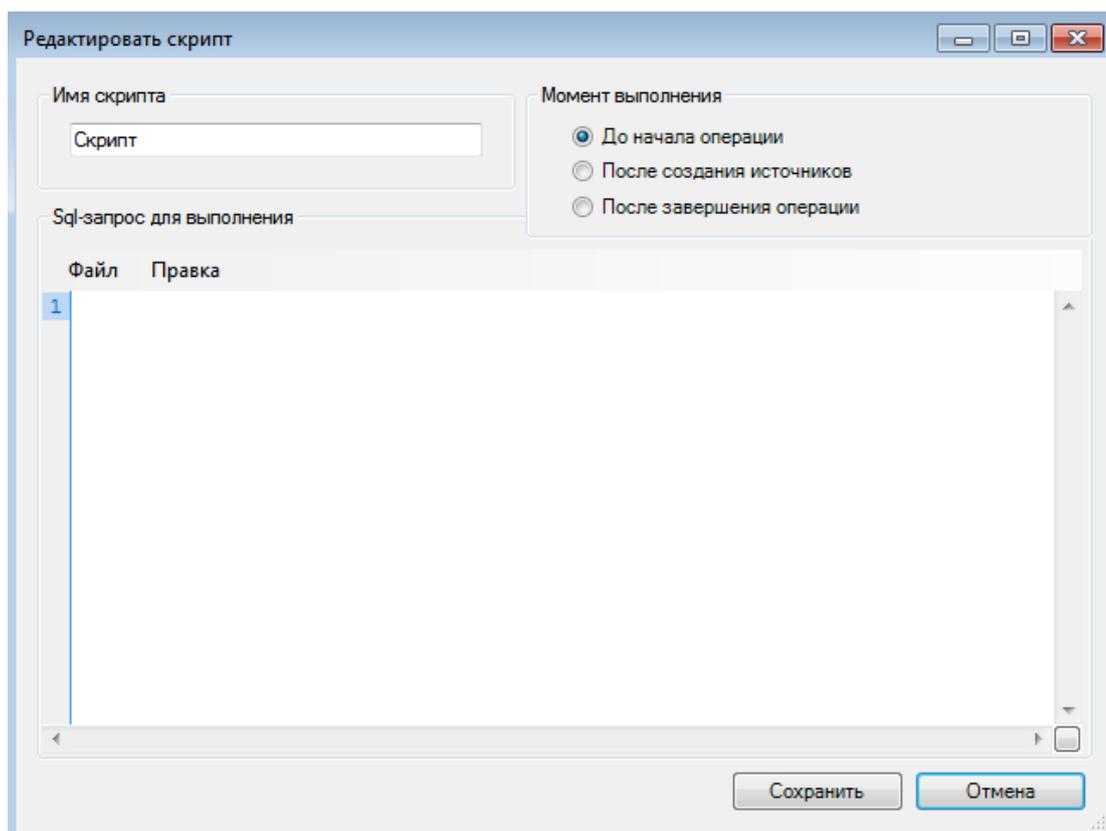


Рисунок 110. Диалог редактирования скриптов

Б.4.3. Отчеты

Отчет – это средство анализа данных. Позволяет предоставить пользователю html-документ с информацией из базы данных, а также определить, необходимо отменить или продолжить работу протокола. Отчет состоит из следующих элементов (Рисунок 111):

- Заголовок отчета – используется в редакторе отчета.
- Переключатель времени выполнения – так же, как и в скрипте, определяет момент выполнения отчета.
- Скрипт настройки параметров отчета.
- Xslt- шаблон для формирования html-отчета.
- SQL-запросы для подстановки данных в xslt- шаблон.

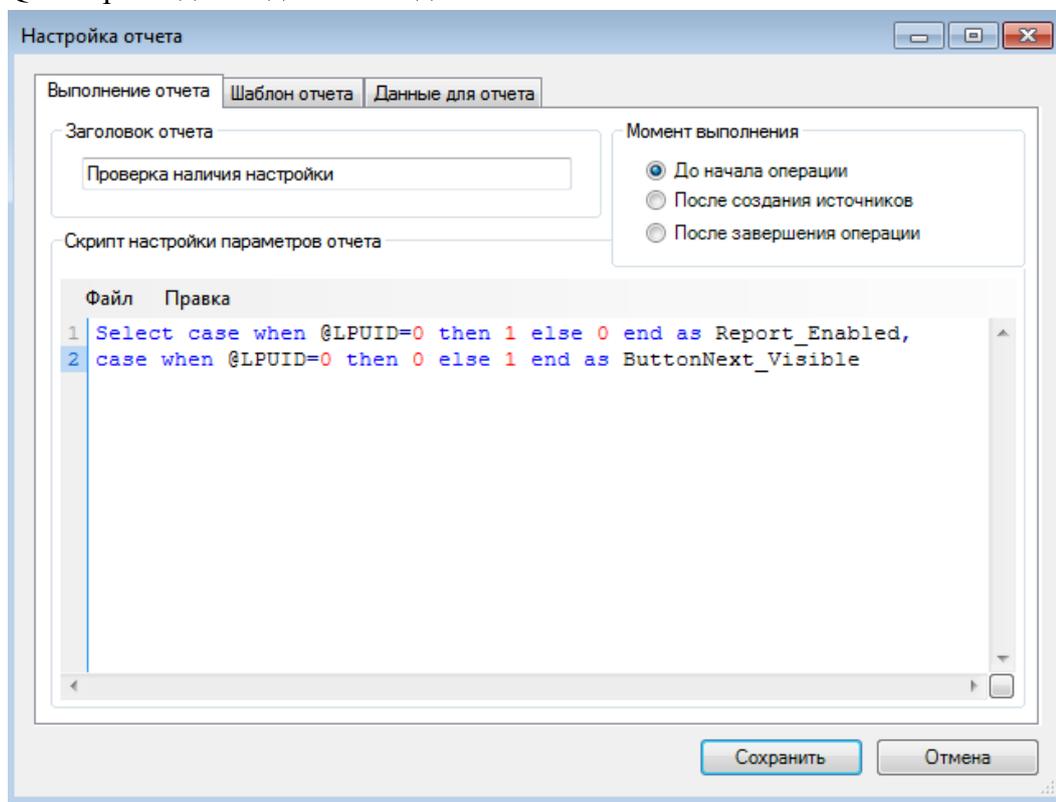


Рисунок 111. Окно «Настройка отчета»

«Скрипт настройки параметров отчета» – это sql-запрос, результаты исполнения которого интерпретируются особым образом. Запрос должен вернуть одну строку с набором полей. В зависимости от имен полей значения в них будут интерпретированы в соответствии с таблицей (Таблица 8).

Таблица 8. Настройки, используемые в отчетах

Имя поля	Альтернативное имя поля	Допустимые значения	Интерпретация
ReportEnabled	Report_Enabled	1 или 0	Требуется ли формировать данный отчет.
ButtonNextVisible	ButtonNext_Visible	1 или 0	Доступна ли пользователю кнопка продолжения выполнения протокола.
ButtonNextText	ButtonNext_Text	Строка	Текст на кнопке продолжения выполнения протокола.
ButtonExitVisible	ButtonExit_Visible	1 или 0	Доступна ли пользователю кнопка остановки выполнения протокола.
ButtonExitText	ButtonExit_Text	Строка	Текст на кнопке остановки

			выполнения протокола.
Message		Строка	Дополнительный текст подсказки для пользователя.

Внимание! Имена полей не зависят от регистра.

Внимание! Скрипт не обязан возвращать все перечисленные значения. По умолчанию отчет всегда включен, доступны обе кнопки с тестом «Отменить» и «Продолжить» соответственно. Достаточно, если в результате выполнения запроса будет получена только часть параметров или даже не получено ни одного, если устраивают все значения параметров отчета по умолчанию.

Например, выполнение нашего протокола не имеет смысла, когда в таблице tmp_Update нет ни одной записи. В этом случае потребуется убрать из отчета кнопку «Продолжить». Это можно выполнить скриптом вида:

```
if not exists (select * from tmp_Update)
    select 0 as ButtonNextVisible
```

Также, обратите внимание: Параметры выполнения отчета связаны с настройкой протокола «Тихий режим выполнения отчетов». Если необходимо чтобы протокол не приостанавливал свое выполнение на отчетах, необходимо выставить «Тихий режим выполнения отчетов» и во всех отчетах реализовать скрипт настройки таким образом, чтобы была доступна только одна из кнопок – «Отмена» или «Продолжить». В этом случае процесс выполнения будет автоматически продолжен или прерван в зависимости от того, какой вариант доступен.

Непосредственно html-отчет, выдаваемый пользователю в процессе выполнения, создается на основе Xslt-шаблона (Рисунок 112).

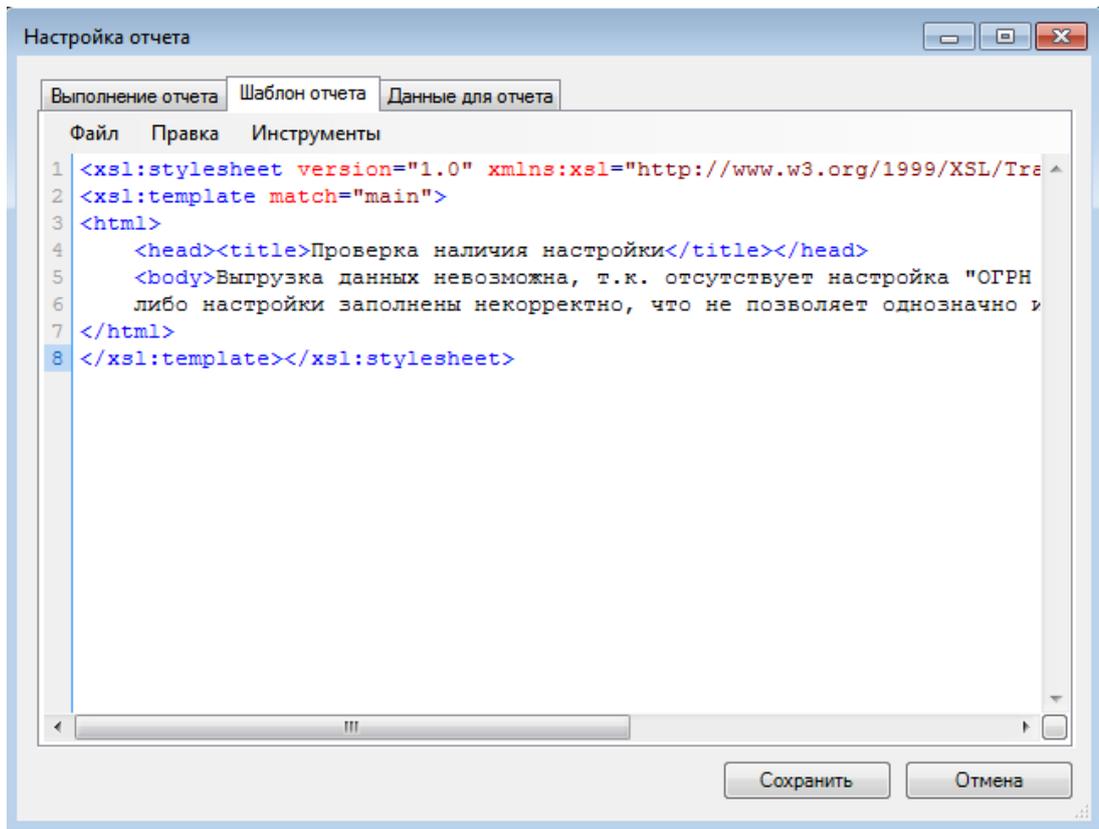


Рисунок 112. Диалог настройки отчета – вкладка «Шаблон отчета»

В шаблон подставляются данные, выбираемые набором sql-запросов (Рисунок 113). Формирование отчета выполняется с помощью так называемой xslt – трансформации. Описание данного механизма выходит за рамки данного документа, поэтому рекомендуется изучить его в специализированной литературе.

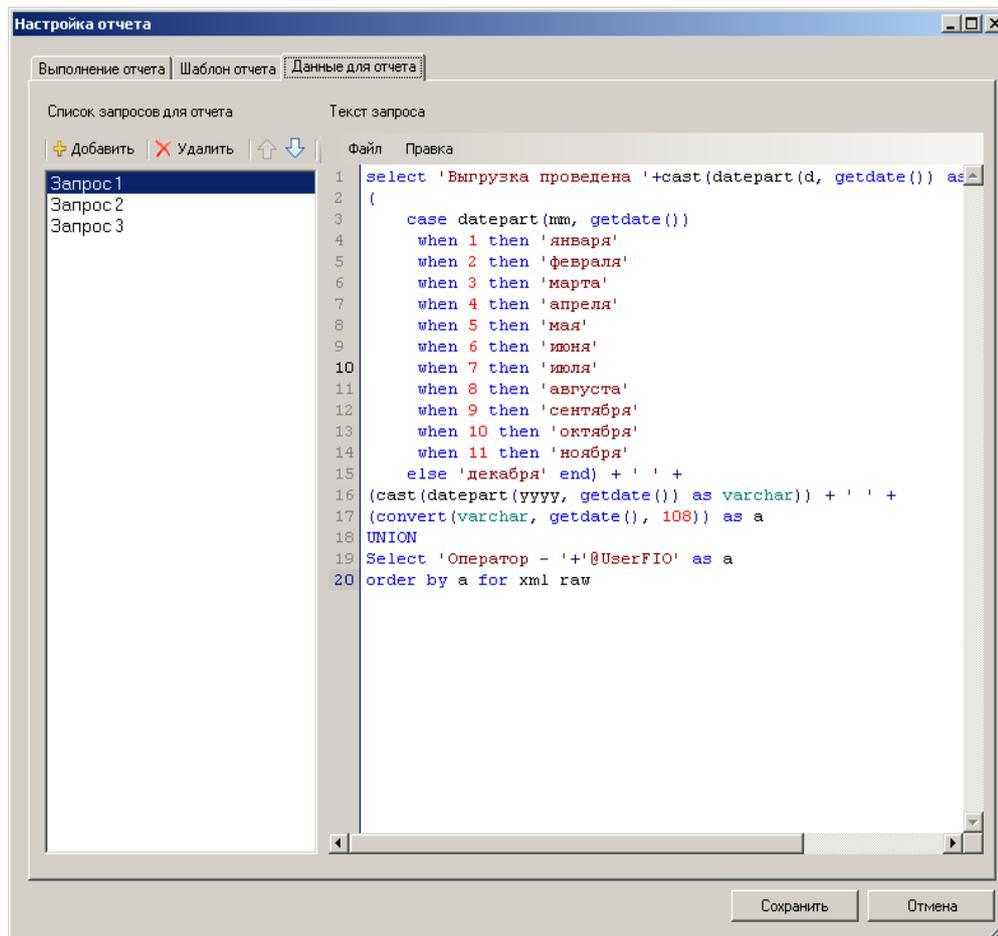


Рисунок 113. Диалог настройки отчета – вкладка «Данные для отчета»

Б.4.4. Источник MSSQL

Источник MSSQL – это набор sql-запросов для формирования временных представлений по содержимому базы данных. Данный источник служит для того чтобы представить существующие данные в виде, наиболее удобном для последующей обработки.

Диалог редактирования источника позволяет создать любое число временных представлений. Для создания представления необходимо указать его имя и sql-запрос выборки данных (Рисунок 114).

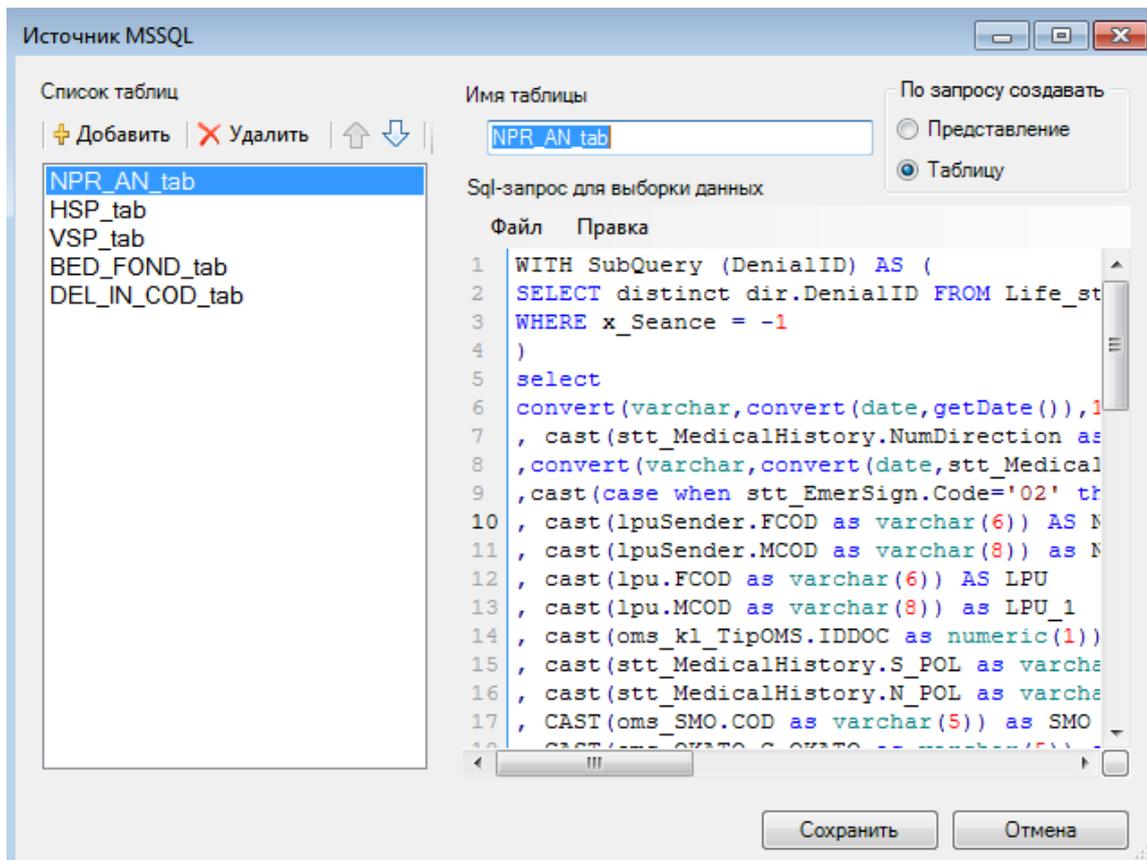


Рисунок 114. Диалог настройки источника MSSQL

Б.4.5. Источник XML

Источник XML – это правило для загрузки xml-файла в одну или несколько временных таблиц. Загрузка данных выполняется средствами библиотеки SqlXml, с помощью XSD-схемы. Обратите внимание, в протоколе может существовать только один файловый источник, т.е. совместно с XML-источником в протоколе может существовать только MSSQL источник.

Диалог настройки источника состоит из трех вкладок (Рисунок 115). Первая вкладка «Файлы» позволяет задать путь, по которому будет осуществлен автоматический поиск файлов для загрузки, а также путь, по которому файлы будут перемещены после удачной загрузки.

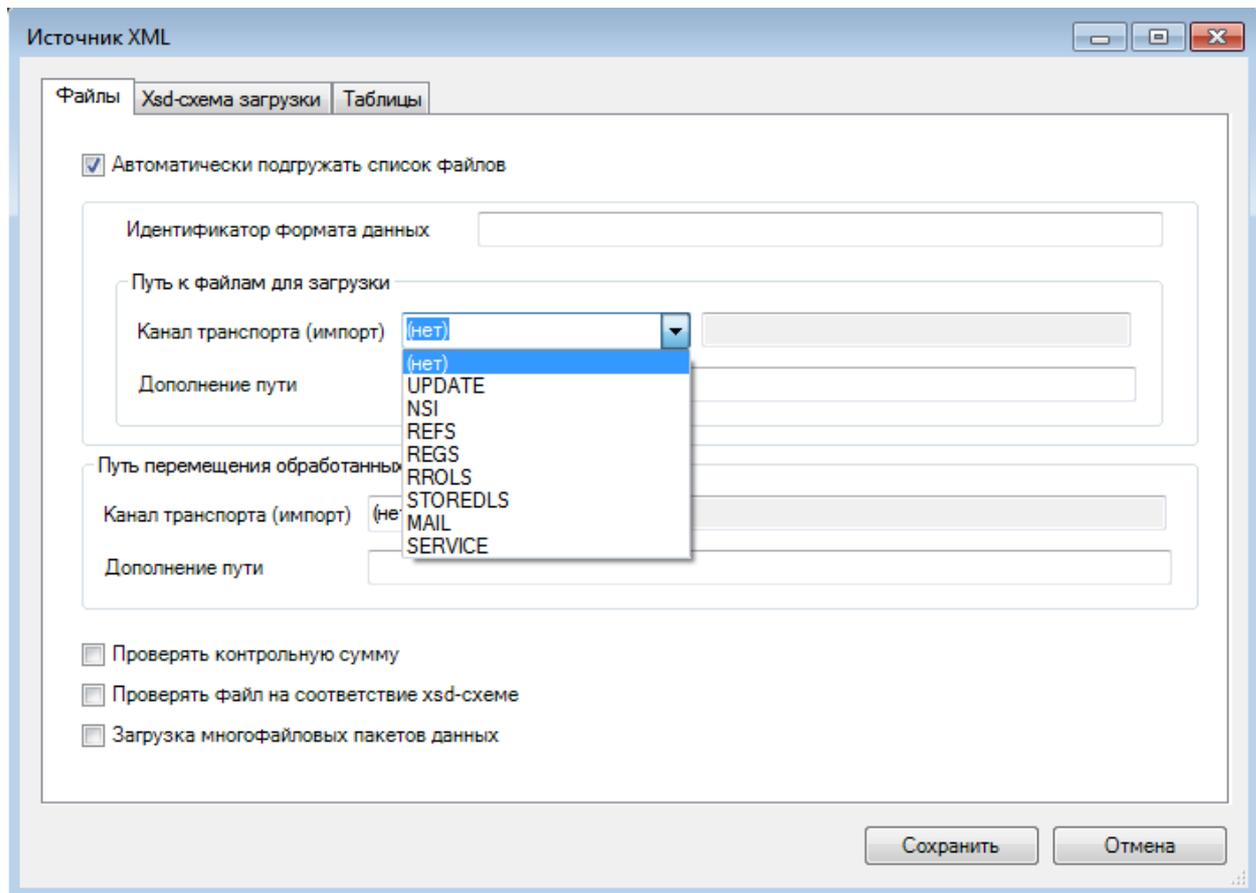


Рисунок 115. Диалог настройки источника XML

Для того чтобы протокол при старте автоматически выбирал доступные для загрузки файлы, необходимо установить флажок «Автоматически подгружать список файлов», и указать «Идентификатор формата данных» и «Путь к файлам для загрузки».

После установки флага «Автоматически подгружать список файлов» при запуске протокола в список файлов автоматически будут добавлены XML-файлы (или ZIP, если они содержат XML). Для поиска файлов будет использован «Путь к файлам для загрузки». При этом XML файл должен содержать тег `FORMAT_GUID` с содержимым, равным тому, что указано в поле «Идентификатор формата данных». Такой механизм позволяет для загрузки, например, электронных накладных просто запустить протокол и сразу же получить все доступные для загрузки файлы.

Путь к файлам для загрузки формируется следующим образом:

- Если в поле «Канал транспорта (импорт)» указан канал импорта, он будет использован в качестве базового пути, если не указан – в качестве базового пути будет использован путь к папке с программой.
- Если в поле «Дополнение пути» указано дополнение пути – оно будет дописано к базовому пути. В дополнении пути можно использовать параметры протокола.
- Если канал импорта не указан, дополнение пути может содержать абсолютный путь.

При старте протокола вычисляется путь загрузки и используется для поиска файлов по их именам. Если файлы будут обнаружены – они будут автоматически выбраны в диалоге загрузки (Рисунок 116).

При удачном завершении загрузки обработанные файлы могут быть перемещены. Для этого необходимо указать соответствующее Дополнение пути. Если дополнение пути содержит относительный путь – файл будет перемещен относительно исходного

местоположения. Если будет указан абсолютный путь – перемещение будет выполнено по заданному пути.

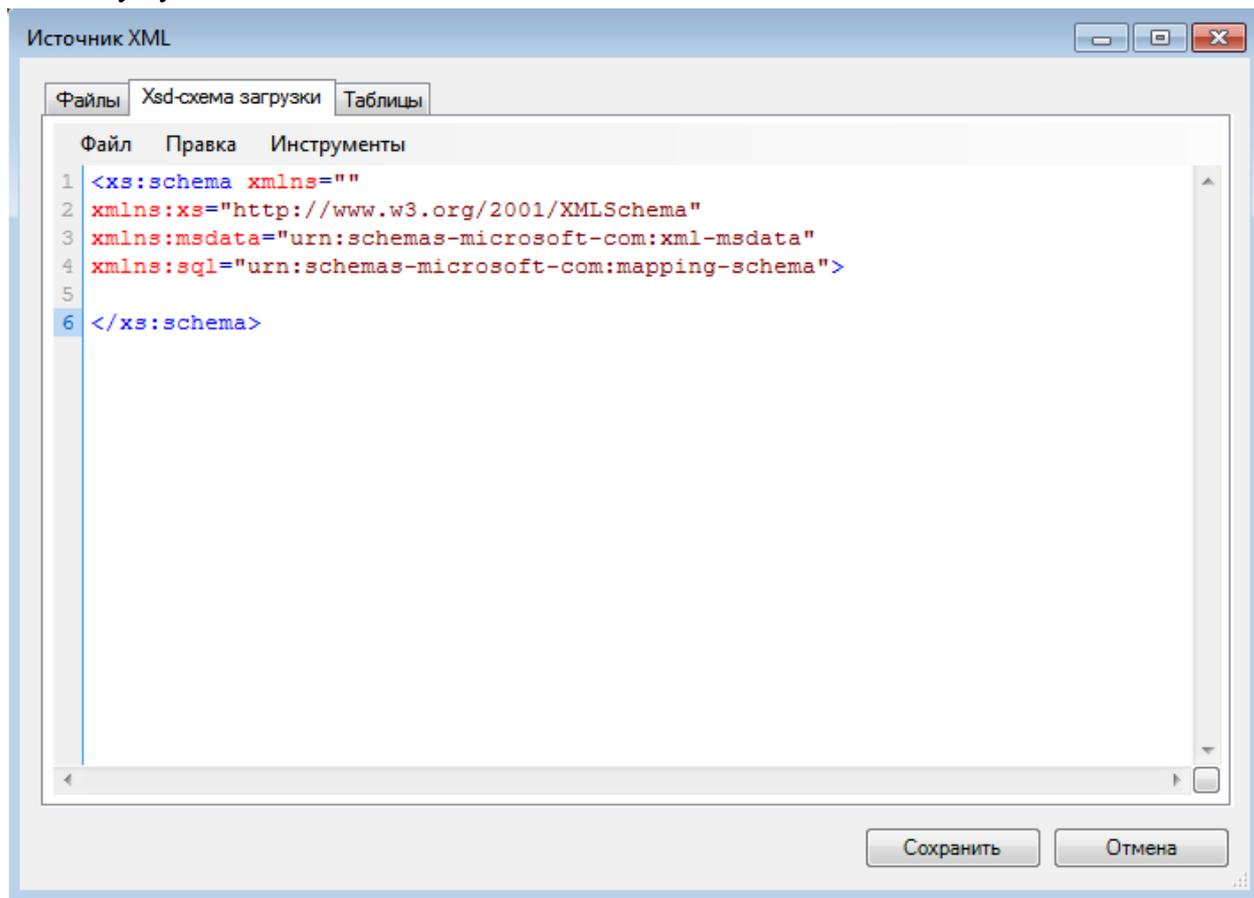


Рисунок 116. Диалог настройки источника XML – вкладка «Xsd – схема загрузки»

Источник XML загружает данные из файлов формата Xml, используя XSD схему с указанными в ней ссылками на временные таблицы, которые будут созданы в базе данных. Таким образом, источник Xml создает временные таблицы. Для начала нужно написать XSD-схему, описывающую структуру загружаемых файлов формата Xml. XSD-схема пишется разработчиком протокола либо предоставляется ему в техническом задании.

После написания стандартной XSD-схемы, описывающей структуру XML-файла, в нее вписываются дополнительные элементы из пространства имен «urn:schemas-microsoft-com:mapping-schema», указывающие, в какие таблицы и в каком формате следует загружать данные.

Пример XSD-схемы с дополнительными элементами, выделенными красным цветом:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sql="urn:schemas-microsoft-com:mapping-schema">
  <xs:annotation>
    <xs:appinfo>
      <sql:relationship name="MAIN_SENDINFO" parent="tmp_MAIN" parent-key="FORMAT_GUID" child="tmp_SENDINFO"
child-key="FORMAT_GUID" />
      <sql:relationship name="MAIN_PERSONDLO" parent="tmp_MAIN" parent-key="FORMAT_GUID"
child="tmp_PERSONDLO" child-key="FORMAT_GUID" />
    </xs:appinfo>
  </xs:annotation>
  <xs:simpleType name="money2">
    <xs:restriction base="xs:decimal">
      <xs:fractionDigits value="2" />
    </xs:restriction>
  </xs:simpleType>
```

```

<xs:simpleType name="recType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="I" />
    <xs:enumeration value="U" />
    <xs:enumeration value="D" />
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="docFlowInfoType">
  <xs:sequence>
    <xs:element name="HOST_GUID" type="xs:string" minOccurs="1" maxOccurs="1" sql:field="OGRN" />
    <xs:element name="TARGET_HOST_GUID" type="xs:string" minOccurs="0" maxOccurs="1" sql:mapped="false" />
    <xs:element name="SEND_GUID" type="xs:string" minOccurs="1" maxOccurs="1" sql:field="GUID" />
    <xs:element name="PREV_SEND_GUID" type="xs:string" minOccurs="0" maxOccurs="1" sql:mapped="false" />
    <xs:element name="FILE_NUMBER" type="xs:integer" minOccurs="0" maxOccurs="1" sql:mapped="false" />
    <xs:element name="PREV_FILE_NUMBER" type="xs:integer" minOccurs="0" maxOccurs="1" sql:mapped="false" />
    <xs:element name="NEXT_FILE_NUMBER" type="xs:integer" minOccurs="0" maxOccurs="1" sql:mapped="false" />
    <xs:element name="PACKAGE_NUMBER" type="xs:integer" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
<xs:element name="MAIN" sql:relation="tmp_MAIN">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="FORMAT_GUID" type="xs:string" minOccurs="1" maxOccurs="1" fixed="{E619D0D5-7430-4840-9E35-C15BC1EF0E3D}" sql:mapped="false" />
      <xs:element name="PROTOCOL" type="xs:string" minOccurs="1" maxOccurs="1" fixed="PHARMACY_REESTR" sql:mapped="false" />
      <xs:element name="VER" type="xs:string" minOccurs="1" maxOccurs="1" fixed="3.0" sql:mapped="false" />
      <xs:element name="CREATE_BY" type="xs:string" minOccurs="0" maxOccurs="1" sql:mapped="false" />
      <xs:element name="APP_BUILD" type="xs:string" minOccurs="0" maxOccurs="1" sql:mapped="false" />
      <xs:element name="CREATE_TIME" type="xs:dateTime" minOccurs="1" maxOccurs="1" sql:datatype="datetime" />
      <xs:element name="TITLE" type="xs:string" minOccurs="0" maxOccurs="1" sql:mapped="false" />
      <xs:element name="ECP" type="xs:string" minOccurs="0" maxOccurs="1" sql:mapped="false" />
      <xs:element name="SENDINFO" type="docFlowInfoType" minOccurs="1" maxOccurs="1" sql:relation="tmp_SENDINFO" sql:relationship="MAIN_SENDINFO" />
      <xs:element name="DATAMAIN" minOccurs="1" maxOccurs="1" sql:is-constant="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="DOCUMENTS" minOccurs="1" maxOccurs="1" sql:is-constant="1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="PERSONDLO_DOC" minOccurs="0" maxOccurs="1" sql:is-constant="1">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="PERSONDLO" minOccurs="0" maxOccurs="unbounded" sql:relation="tmp_PERSONDLO" sql:relationship="MAIN_PERSONDLO">
                          <xs:complexType>
                            <xs:sequence>
                              <xs:element name="SS" type="xs:string" minOccurs="1" maxOccurs="1" />
                              <xs:element name="S_POL" type="xs:string" minOccurs="1" maxOccurs="1" />
                            </xs:sequence>
                            <xs:attribute name="op" type="recType" use="required" sql:mapped="true" />
                          </xs:complexType>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:attribute name="chsm" type="xs:string" use="required" sql:mapped="false" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

Xml документ, соответствующий приведенной схеме:

```

<?xml version="1.0" encoding="windows-1251"?>
<MAIN chsm="909FC318A5F9708B1592D2851D81062CCD845D6C">
  <FORMAT_GUID>{E619D0D5-7430-4840-9E35-C15BC1EF0E3D}</FORMAT_GUID>
  <PROTOCOL>PHARMACY_REESTR</PROTOCOL>
  <VER>3.0</VER>
  <CREATE_BY>Library</CREATE_BY>
  <CREATE_TIME>2007-03-06T16:41:44</CREATE_TIME>
  <TITLE>Обновления справочников</TITLE>
  <SENDINFO>
    <HOST_GUID>000</HOST_GUID>
    <SEND_GUID>{e8bde092-dd1b-4f98-8447-6b1f3ff183e3}</SEND_GUID>
    <FILE_NUMBER>21</FILE_NUMBER>
    <PREV_FILE_NUMBER>20</PREV_FILE_NUMBER>
    <NEXT_FILE_NUMBER>22</NEXT_FILE_NUMBER>
    <PACKAGE_NUMBER>1</PACKAGE_NUMBER>
  </SENDINFO>
  <DATAMAIN>
    <DOCUMENTS>
      <PERSONDLO_DOC>
        <PERSONDLO op="I">
          <SS>111-222-333 44</SS>
          <S_POL>1234</S_POL>
        </PERSONDLO>
        <PERSONDLO op="I">
          <SS>444-333-222 11</SS>
          <S_POL>987654321</S_POL>
        </PERSONDLO>
      </PERSONDLO_DOC>
    </DOCUMENTS>
  </DATAMAIN>
</MAIN>

```

Комментарии к XSD:

xmlns:sql="urn:schemas-microsoft-com:mapping-schema" – использовать пространство имен urn:schemas-microsoft-com:mapping-schema под именем sql – это пишется во всех схемах;

<sql:relationship name="MAIN_SENDINFO" parent="tmp_MAIN" parent-key="FORMAT_GUID" child="tmp_SENDINFO" child-key="FORMAT_GUID" /> – внутри этого тега описывается связь между данными из родительского тега, загружаемыми в одну таблицу и данными, загружаемыми в другую таблицу из вложенных тегов. Эта связь описывает, какие поля из родительского тега будут включены в таблицу данных, загружаемых из вложенных тегов. В примере описана связь с названием **MAIN_SENDINFO**, между таблицей **tmp_MAIN**, загружаемой из родительского тега, и таблицей **tmp_SENDINFO**, загружаемой из дочерних (вложенных) тегов. Поле, которое добавляется в таблицу **tmp_SENDINFO** - **FORMAT_GUID**. Если связь между таблицами должна идти по нескольким полям, то в значении **child-key** они пишутся через пробел;

sql:field="OGRN" – если поле в создаваемой таблице должно называться иначе, чем тег с данными, то можно указать имя поля;

sql:mapped="false" – указывается, что данные из этого поля не должны выгружаться в таблицы;

sql:relation="tmp_MAIN"> – указывается имя создаваемой временной таблицы. Все атрибуты тега и дочерние теги с данными будут выгружены в эту таблицу, кроме отмеченных как **sql:mapped="false"**. Если в теге, в который вложен данный тег, уже встретился атрибут **sql:relation**, то в данный тег необходимо добавить атрибут **sql:relationship**, который содержит имя связи таблиц, описанной ранее;

sql:relationship="MAIN_SENDINFO" – название связи, описанной в **sql:relationship**, указывается для таблицы, загружаемой из дочерних элементов;

sql:is-constant="1" – если данный тег не содержит атрибутов и вложенных тегов с данными, и не должен учитываться при загрузке данных в таблицы, то для него в XSD-схеме указывается этот атрибут;

sql:datatype – тип поля в таблице, в которое загрузятся данные из данного элемента. Если этот атрибут отсутствует, то тип данных выбирается автоматически, исходя из типа элемента. Если тип элемента “xs:date” или “xs:dateTime”, то необходимо указать атрибут **sql:datatype="datetime"**.

При загрузке данного примера Xml по данной схеме будут созданы следующие таблицы (верхняя строка – названия полей таблицы):

Таблица **tmp_MAIN**

CREATE_TIME
2007-03-06T16:41:44

Таблица **tmp_SENDINFO**

OGRN	GUID
000	{e8bde092-dd1b-4f98-8447-6b1f3ff183e3}

Таблица **tmp_PERSONDLO**

SS	S_POL
111-222-333 44	1234
444-333-222 11	987654321

Внимание! Может возникнуть ситуация, когда при описании схемы не будет выявлено явных ключевых полей для таблицы при описании связки **<sql:relationship>**. Тогда можно создать ключевое поле для родительской таблицы. После этого надо прописать в XSD схеме в родительском теге, у которого не найдено дочерних тегов, которые можно объявить ключевыми полями в связке, дочерний тег с таким же именем, как и это поле, со следующими атрибутами: **type="xs:decimal" minOccurs="0" maxOccurs="0" sql:datatype="int"**. Далее можно использовать это поле для задания ключа в **<sql:relationship>**. Что произойдет: в родительской таблице будет создано поле с указанным именем типа **int**, **identity** – автоинкрементное поле-идентификатор. Соответственно оно автоматически заполнится в родительской таблице и будет так же в дочерней. Таким образом, в дочерней таблице будет явная ссылка на родительскую таблицу (Рисунок 117).

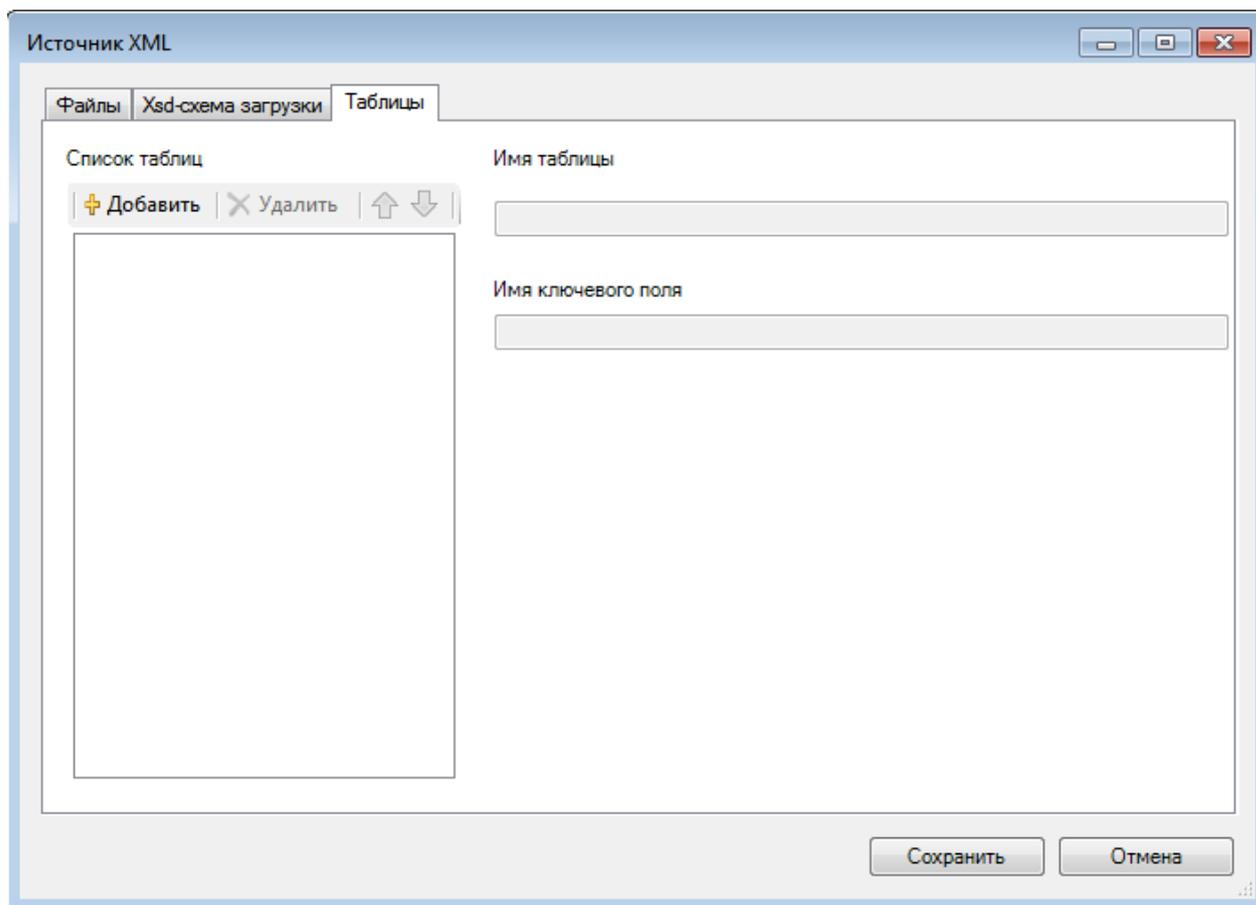


Рисунок 117. Диалог настройки источника XML – вкладка «Таблицы»

Б.4.6. Источник DBF

Источник DBF – это средство загрузки DBF файлов во временные таблицы. Один источник позволяет загружать несколько файлов, при этом каждый файл загружается в отдельную таблицу. Обратите внимание, в протоколе может существовать только один файловый источник, т.е. совместно с DBF-источником в протоколе может существовать только MSSQL источник.

Диалог настройки источника состоит из 2х вкладок. Первая вкладка «Пути к файлам» позволяет задать путь, по которому будет осуществлен поиск файлов для загрузки, а так же путь, по которому файлы будут перемещены после удачной загрузки (Рисунок 118).

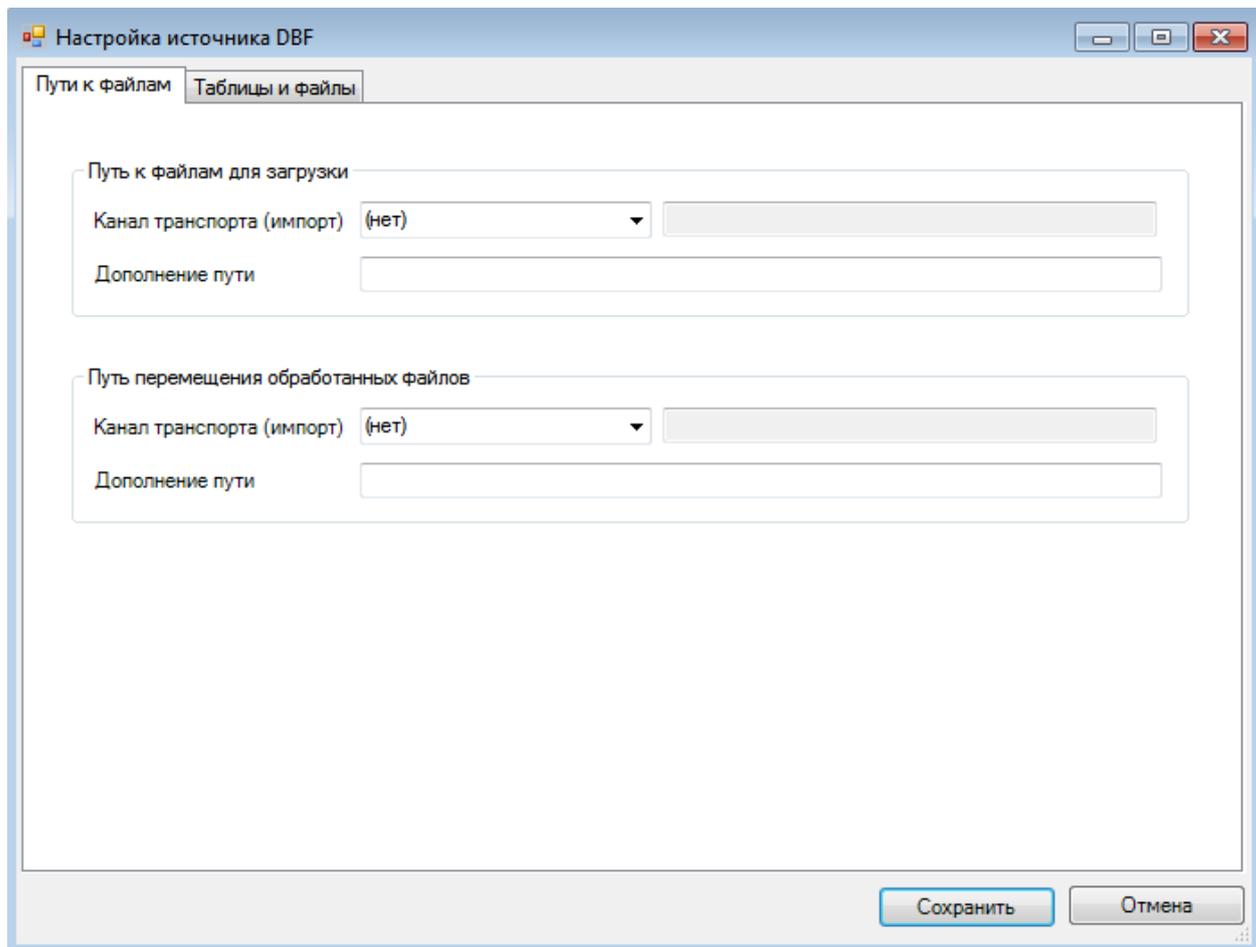


Рисунок 118. Диалог настройка источника DBF

Путь к файлам для загрузки работает следующим образом:

- Если указан канал импорта, он будет использован в качестве базового пути, если не указан – в качестве базового пути будет использован путь к папке с программой.
- Если заполнено поле «Дополнение пути», оно будет дописано к базовому пути. В дополнении пути можно использовать параметры протокола.
- Если канал импорта не указан, поле «Дополнение пути» может содержать абсолютный путь.

При старте протокола вычисляется путь загрузки и используется для поиска файлов по их именам. Если файлы будут обнаружены – они будут автоматически выбраны в диалоге загрузки.

При удачном завершении загрузки обработанные файлы могут быть перемещены. Для этого необходимо указать соответствующее Дополнение пути. Если дополнение пути содержит относительный путь – файл будет перемещен относительно исходного местоположения. Если будет указан абсолютный путь – перемещение будет выполнено по заданному пути.

На второй вкладке диалога настройки задается перечень загружаемых файлов, соответствующие им временные таблицы, а также способ загрузки каждого из них (Рисунок 119). Для каждой из временных таблиц задается:

- Рекомендуемое имя файла (без расширения).
- Комментарий к файлу, который будет показан пользователю в диалоге выбора.

- Кодировка, используемая при загрузке файла (рекомендуется использовать автоматическое определение).
- Способ загрузки данных из файла – автоматическая загрузка всего файла во временную таблицу или загрузка данных во временную таблицу по sql-запросу.

Внимание! При загрузке данных по sql-запросу, все содержимое файла в любом случае будет загружено в базу данных, а затем уже перенесено запросом во временную таблицу.

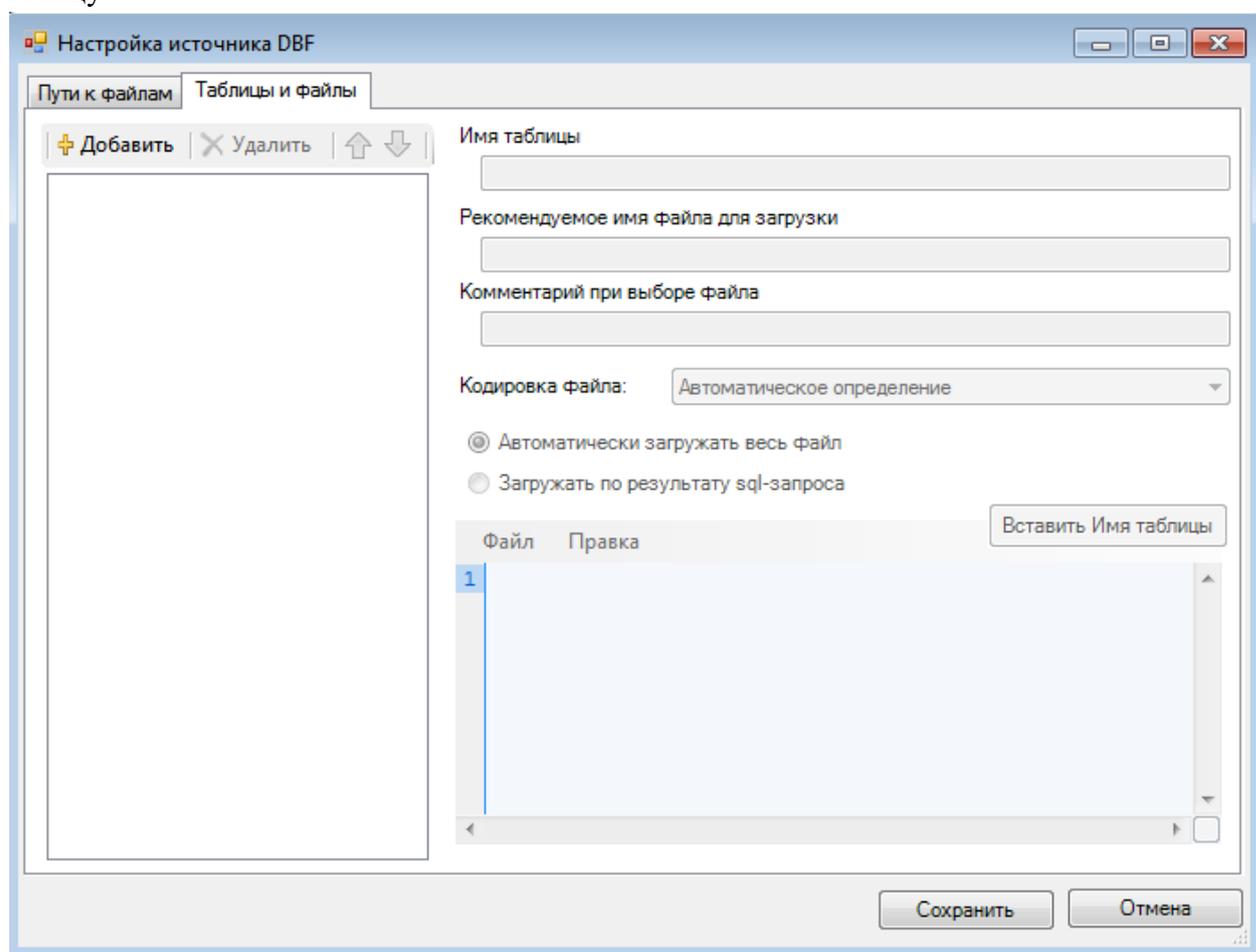


Рисунок 119. Диалог Настройка источника DBF – вкладка «Таблицы и файлы»

Б.4.7. Приемник MSSQL

Приемник MSSQL – это набор правил переноса (загрузки) данных таблицы базы данных из временных (или любых других) таблиц в постоянные таблицы.

В приемнике требуется объявить список таблиц, в которые будет выполняться перенос данных (Рисунок 120). Для каждой из таблиц нужно указать, откуда будет выполнена выборка данных, а также метод переноса. Для выборки допустимо указывать постоянные таблицы, временные таблицы или sql-запрос.

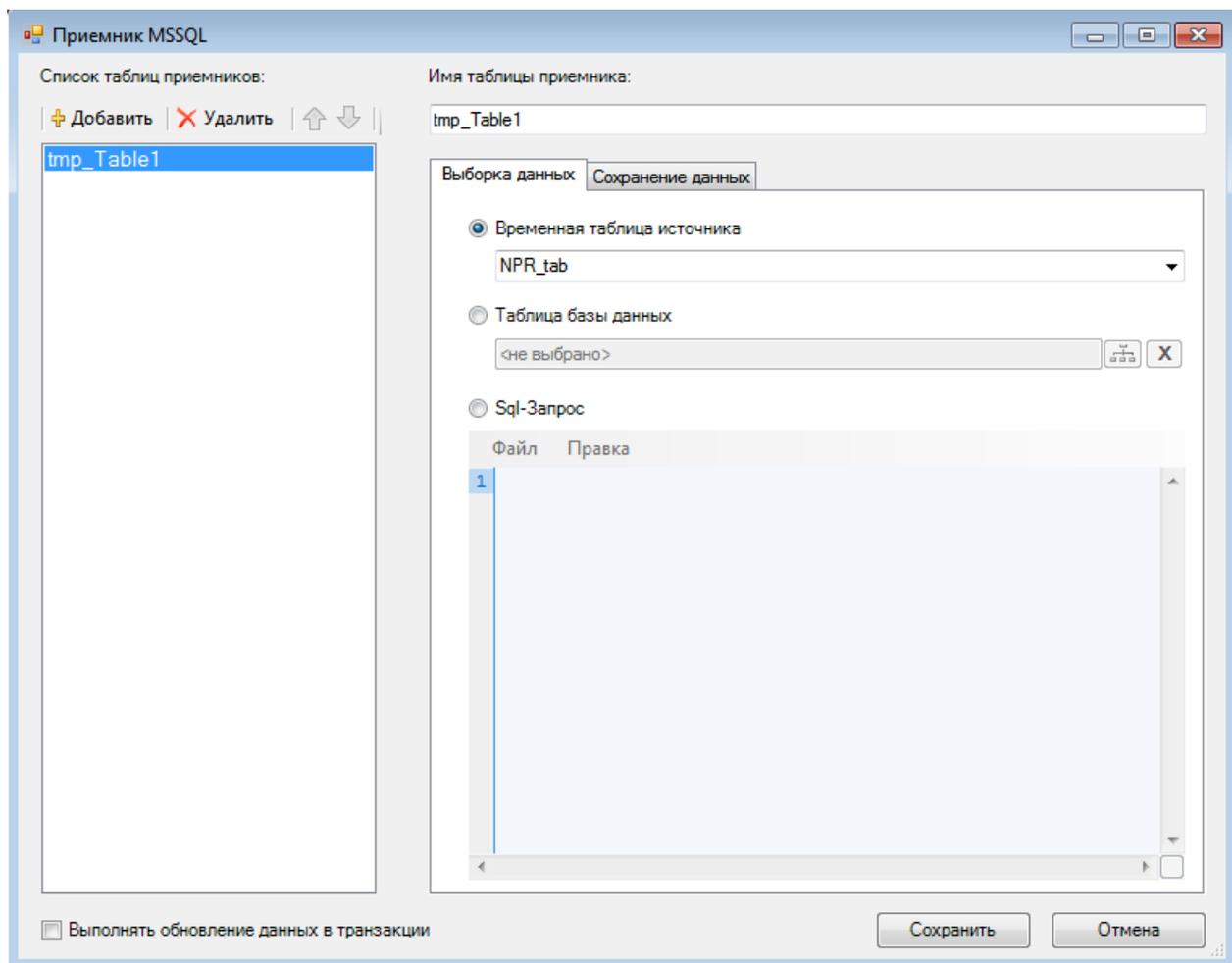


Рисунок 120. Диалог настройки приемника MSSQL – Выборка данных

В качестве метода переноса допустимо (Рисунок 121):

- Добавление – все данные будут добавлены в таблицу.
- Добавление и обновление – на основе данных о ключевых полях, новые записи будут добавлены, а уже существующие – обновлены.
- Удаление – на основе данных о ключевых полях в таблице будут удалены все те записи, которые существуют в выборке.
- Обновление – на основе данных о ключевых полях будет выполнена попытка обновить записи соответствующими им данными из выборки.

Если метод переноса требует сведений о ключевых полях, их требуется обязательно указать. Также возможно на момент работы приемника включить для таблицы вставку в автоинкрементные столбцы (Identity).

Вся работа приемника MSSQL может выполняться в одной транзакции, однако это не будет происходить, если в самом протоке объявлено выполнение в транзакции.

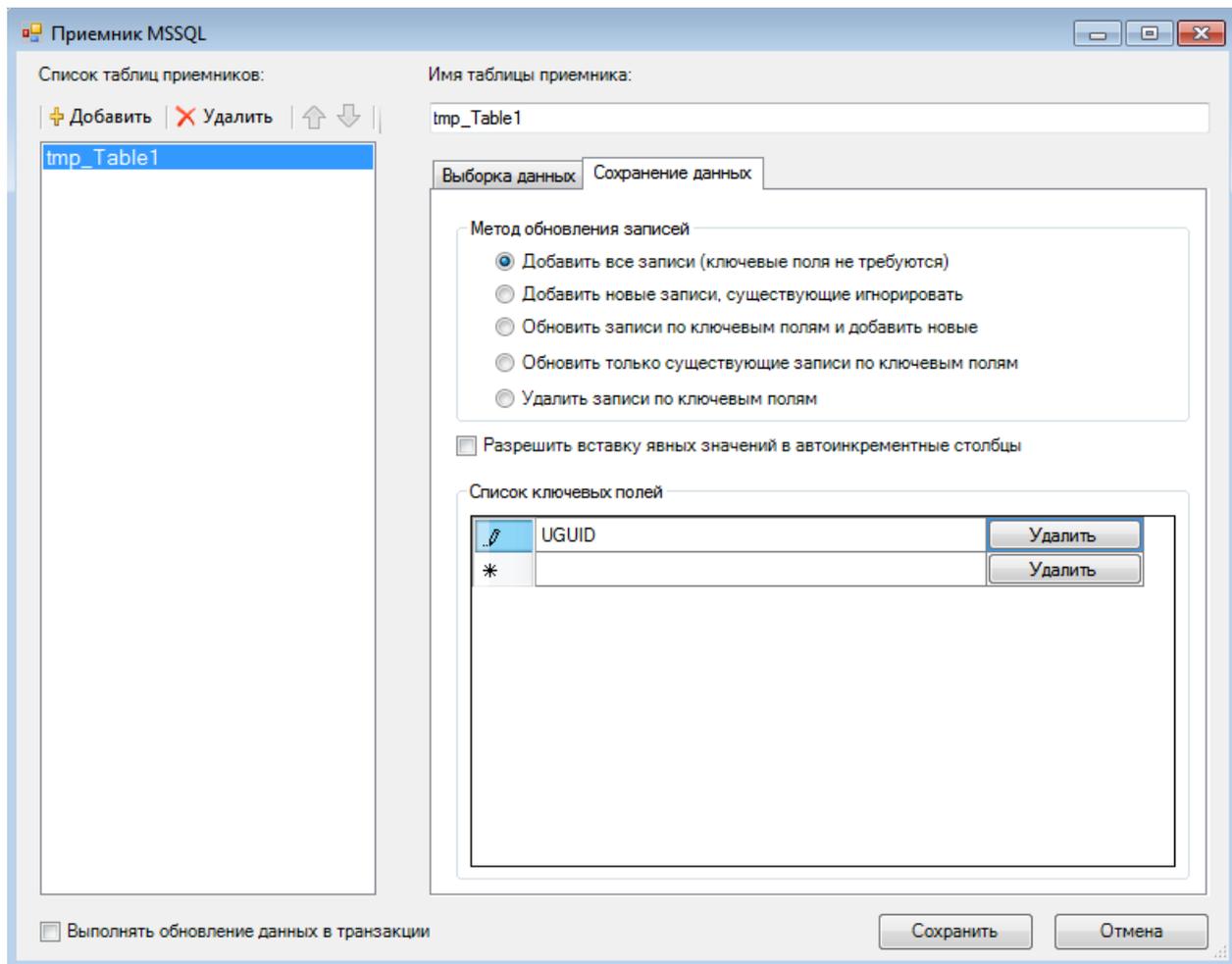


Рисунок 121. Диалог настройки приемника MSSQL – Обновление данных

Б.4.8. Приемник XML

Приемник XML – это правило выгрузки данных в файлы формата XML. При необходимости, приемник может разбить полученный Xml-файл на файлы определенного размера. В зависимости от настроек с каждым из полученных файлов производятся следующие операции: дописывание заголовков, вычисление и запись контрольной суммы, Xslt-преобразование, архивирование. Если вам необходимо выгружать сразу несколько различных файлов формата Xml, то добавьте несколько приемников Xml в протокол.

Диалог настройки приемника XML позволяет задавать имя и путь выгрузки файлов (Рисунок 122). При этом допустимо использовать следующие подстановки:

- [LibName] – имя библиотеки-создателя;
- [CreateTime] – время создания файла;
- [GenGUID] – сгенерировать новый GUID;
- [Sql xyz] – поместить в поле результат запрос XYZ;
- [SendGuid] – GUID посылки;
- [PrevSendGuid] – GUID предыдущей посылки по данному протоколу;
- [FileNum] – номер файла посылки;
- [PrevFileNum] – номер предыдущего файла в посылке;
- [NextFileNum] – номер следующего файла в посылке;
- [PakNum] – номер посылки.

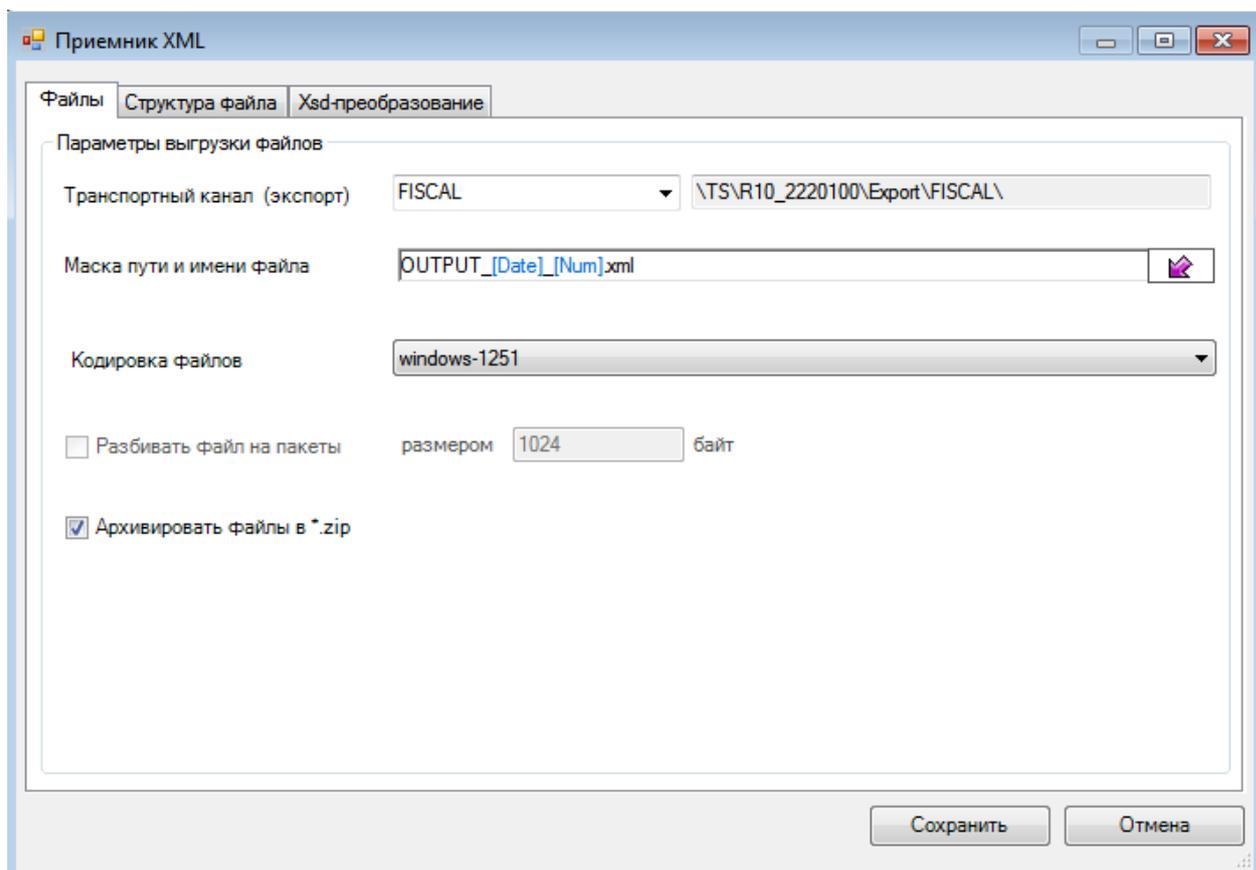


Рисунок 122. Диалог настройки выгрузки XML, вкладка «Файлы»

На вкладке «Файлы» нужно указать параметры выгрузки файлов: транспортный канал, маску пути и имени файла, его кодировку. Флаг «Архивировать файлы в *.zip» позволяет получать на выходе архивы, содержащие выходной XML. При установке флага «Разбить файл на пакеты» будет включено разбиение выходного файла на несколько частей.

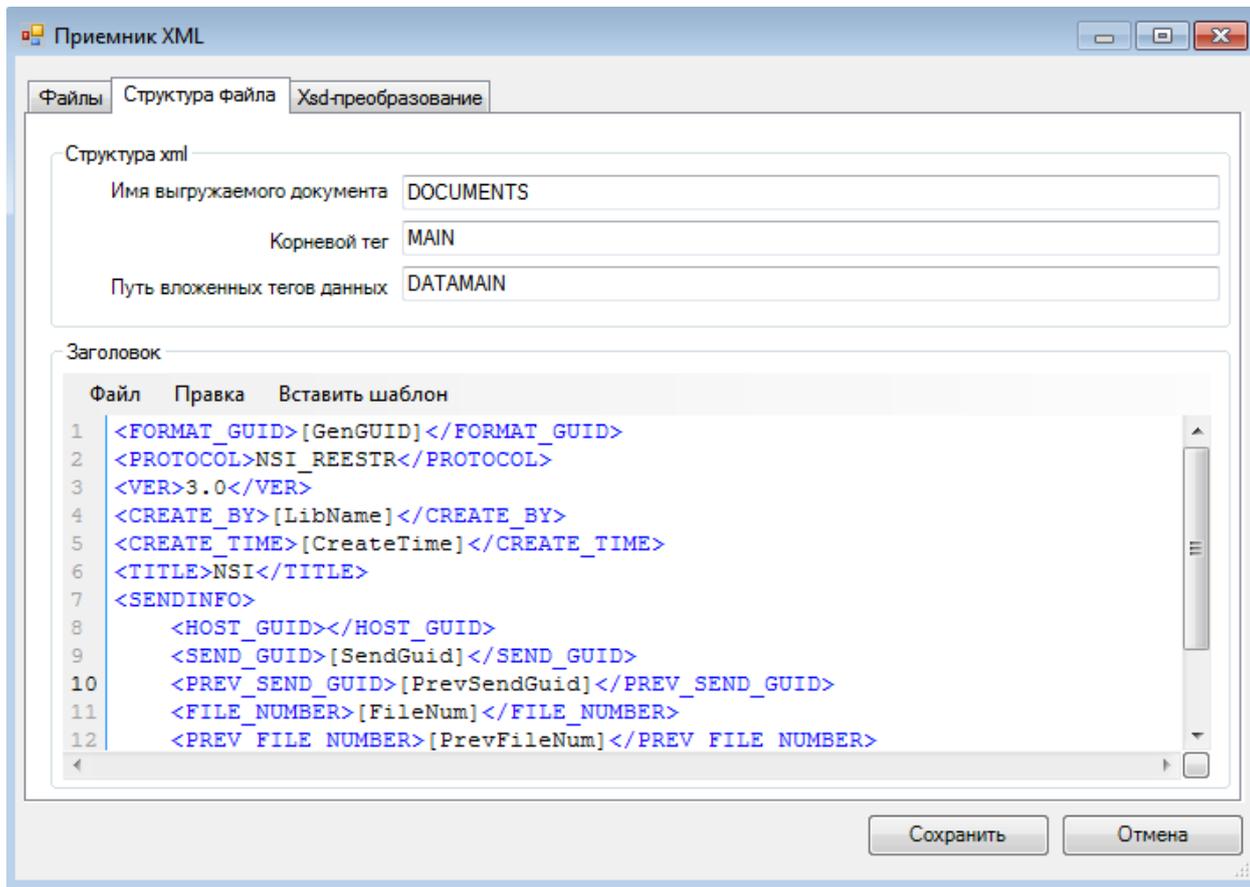


Рисунок 123. Диалог настройки выгрузки XML, вкладка «Структура файла»

На вкладке «Структура файла» отображается информация о структуре выгружаемого документа. Имя выгружаемого документа должно соответствовать корневому тегу XSD-схемы. Главный тег – это имя корневого тега в выгружаемых файлах. Путь вложенных тегов данных служит для упрощения создания схемы, он указывает теги, в которые будут вкладываться теги, описанные в схеме. Например, пусть нужно получить документ:

```

<DATAMAIN>
  <DOCUMENTS>
    <PERSONDLO_DOC>
      <PERSONDLO>
        <SS>111-111-111 11</SS>
        <S_POL>С</S_POL>
        <N_POL>1111</N_POL>
        <FAM>Злобнина</FAM>
        <IM>Ольга</IM>
        <OT>Ивановна</OT>
      </PERSONDLO>
    </PERSONDLO_DOC>
  </DOCUMENTS>
</DATAMAIN>

```

Тогда достаточно будет описать в схеме тег <PERSONDLO> и указать путь вложенности «DATAMAIN/DOCUMENTS/PERSONDLO_DOC».

В поле «Заголовок файлов» описывается структура заголовка выгружаемых файлов. При этом в заголовке можно использовать различные подстановки, все они доступны при выборе пункта меню «Шаблон» (Рисунок 124).

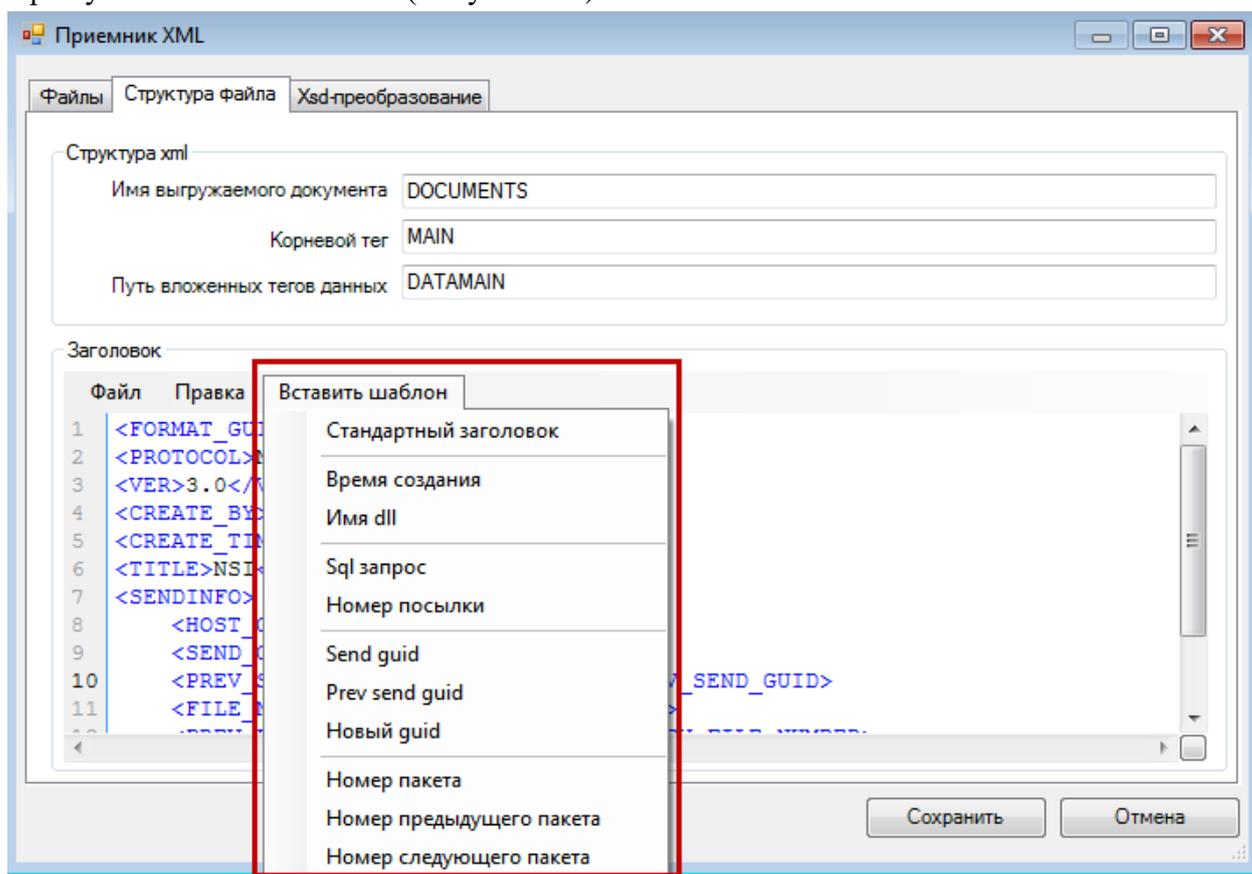


Рисунок 124. Выбор шаблона в поле «Заголовок» на вкладке «Структура файла» окна настройки выгрузки XML

Создание XSD-схем – самый важный этап при настройке приемника XML. XSD-схемы включают модели содержания элементов. Модель содержания элемента сложного типа – формальное описание структуры и допустимого содержания элемента, которое используется для проверки правильности XML документа. Модель содержания может ограничивать документ до некоторого набора элементных типов и атрибутов, описывать и поддерживать связи между этими различными компонентами и уникально обозначать отдельные элементы. Свободное использование модели содержания позволяет разработчикам изменять структурную информацию.

Любая схема начинается с открывающего тега `schema`, который также содержит объявление используемых пространств имен (Рисунок 125). Любая схема должна использовать пространство имен `xmlns:xsd="http://www.w3.org/2001/XMLSchema"` (возможно переименовать пространство имен, т.е. использовать не пространство `xsd`, а, например, `xs`). Для задания соответствия xml и таблиц базы данных используется пространство имен `xmlns:sql="urn:schemas-microsoft-com:mapping-schema"`. Его также можно переименовать.

Перечень объявлений дочерних элементов приводится в структуре группирующих XSD-элементов `choice`, `sequence`, и `all`. Элемент `xsd:choice` позволяет только одному из элементов, содержащихся в группе, присутствовать в составе элемента. Элемент `xsd:sequence` требует появления элементов группы в точно установленной последовательности в составе

элемента. Элемент `xsd:all` – позволяет элементам в группе быть (или не быть) в любом порядке в составе элемента.

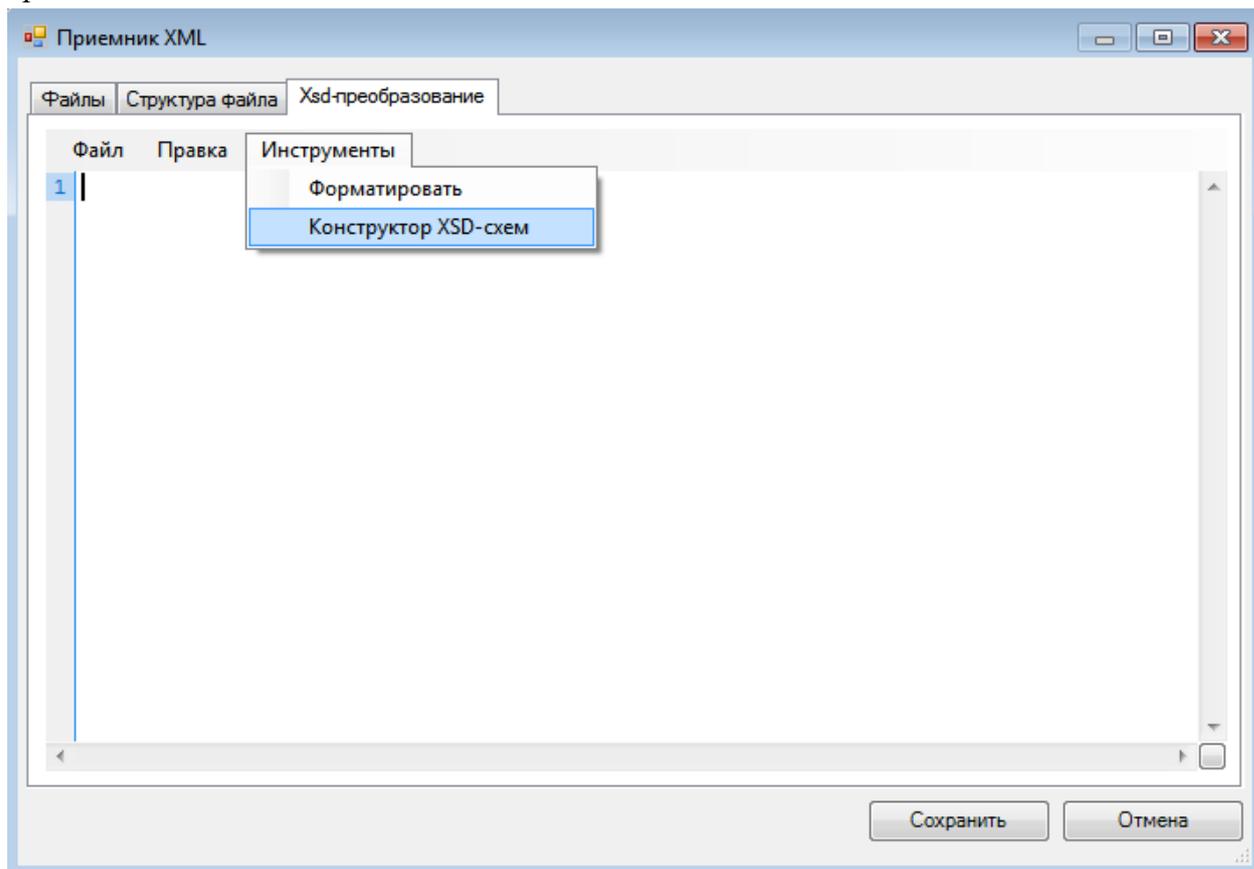


Рисунок 125. . Диалог настройки выгрузчика XML – вкладка «Xsd – преобразование», выбор конструктора

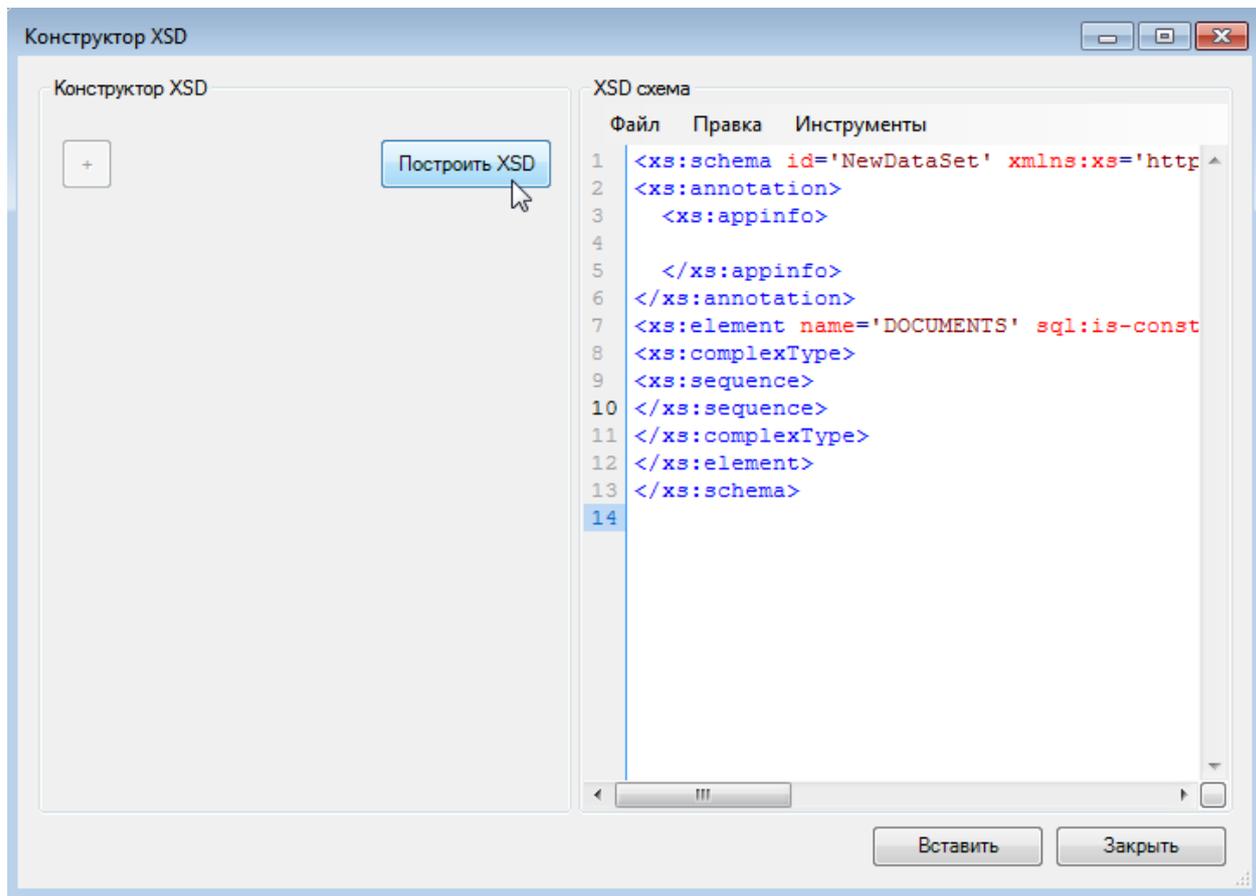


Рисунок 126. Диалог настройки выгрузчика XML – вкладка «Xsd – преобразование», конструктор Xsd

Элементы и атрибуты описываются соответственно тегами *xsd:element* и *xsd:attribute*. Имена элементов и атрибутов в конечном документе задаются атрибутом *name* соответствующего тега схемы.

Любой тег, в который вложены другие, считается сложным типом. Сложные типы описываются с помощью тега *xsd:complexType*, в который вкладываются теги *xsd:sequence* и теги, описывающие атрибуты.

Простейший пример XSD-схемы, описывающей сложный тип:

```

1 <?xml version="1.0" encoding="windows-1251" ?>
2 <xsd:schema id="NewDataSet" xmlns=""
3 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4 <xsd:element name="Main">
5 <xsd:complexType>
6 <xsd:sequence>
7 <xsd:element name="Person">
8 <xsd:complexType>
9 <xsd:sequence>
10 <xsd:element name="Name"/>
11 <xsd:element name="Surname"/>
12 <xsd:element name="DateOfBirth"/>
13 </xsd:sequence>
14 <attribute name="PersonalID"/>
15 </xsd:complexType>
16 </xsd:element>
17 </xsd:sequence>
18 </xsd:complexType>
19 </xsd:element>

```

Разберем данную схему по строкам:

- 1: заголовок XML-документа;
- 2 и 3: начало корневого тега *schema*. Объявляются необходимые пространства имен;
- 4: объявление корневого элемента XML-документа *Main*. *Main* является сложным типом, поэтому он должен включать тег *complexType*;
- 5: начальный тег *complexType*;
- 6: начальный тег *sequence*;
- 7: объявляется составной элемент *Person*, вложенный в *Main*. Он в свою очередь тоже является сложным типом;
- 8 и 9: определение сложного типа и последовательности элементов;
- 10,11 и 12: дочерние элементы, содержащие данные;
- 13: конец последовательности элементов;
- 14: атрибут элемента *Person*;
- 15: конец объявления сложного типа;
- 16-20: закрывающие теги.

Приведем также пример документа, отвечающего данной схеме:

```
<?xml version="1.0" encoding="windows-1251" ?>
<Main>
  <Person PersonalID="59812">
    <Name>Ivan</Name>
    <Surname>Ivanov</Surname>
    <DateOfBirth>10/05/1980</DateOfBirth>
  </Person>
  <Person PersonalID="37890">
    <Name>Sidor</Name>
    <Surname>Sidorov</Surname>
    <DateOfBirth>21/10/1978</DateOfBirth>
  </Person>
</Main>
```

В нашем случае XSD-схемы описывают порядок выгрузки данных из таблиц MSSQL в XML-документ. При этом указывается, из каких таблиц в какие теги и атрибуты должны попадать значения. Соответствие элемента и таблицы представляется атрибутом элемента `sql:relation="ИМЯ_ТАБЛИЦЫ"`. Элементы, представляющие таблицы, всегда являются сложными типами, при этом все включенные в него элементы по умолчанию отображают строки этой таблицы. Соответствие элемента/атрибута и поля таблицы задается атрибутом `sql:field` соответствующего тега схемы. Необходимо также отметить, что ставить соответствие полей таблицы стоит только в том случае, если имена тегов и полей отличаются, в противном случае соответствие устанавливается автоматически.

Элементы, для которых нет соответствия ни таблиц, ни полей, должны быть помечены атрибутом `sql:is-constant="1"`.

Приведем пример, описывающий выгрузку простейших данных в XML из таблицы `oms_Person`:

```

<xsd:schema id="NewDataSet" xmlns=""
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sql="urn:schemas-microsoft-com:mapping-schema">
  <xsd:element name="Main" sql:is-constant="1">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Person" sql:relation="oms_Person">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Name"/>
              <xsd:element name="Surname" sql:field="FAMILY"/>
              <xsd:element name="DateOfBirth" sql:field="DR"/>
            </xsd:sequence>
            <attribute name="PersonalID" sql:field="PersonID"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

Этой схемы вполне достаточно, чтобы выгрузить данные о льготниках (имя, фамилию и дату рождения) из таблицы в XML-файл.

Курсивом в примере отмечены отличия от предыдущего примера. Первое отличие – дополнительное пространство имен, отвечающее за разметку соответствия структуры XML-файла и структуры таблиц БД.

При объявлении тега добавлен атрибут *sql:is-constant="1"*. Это означает, что тег *Main* отвечает только за структуру XML-документа и ему нет соответствия в таблицах. А вот для элемента *Person* отмечено соответствие таблице *oms_Person*. Поскольку элемент, представляющий таблицу, обязательно является сложным типом, далее описаны подчиненные элементы и атрибуты, включенные в элемент *Person*. Для всех элементов, кроме *Name*, описано соответствие полям таблицы, потому что имена элементов документа и столбцов таблицы различаются.

После выполнения данной выгрузки получаются файлы, похожие структурой на файл из первого примера:

```

<?xml version="1.0" encoding="windows-1251" ?>
<Main>
  <Person PersonalID="59812">
    <Name>Ivan</Name>
    <Surname>Ivanov</Surname>
    <DateOfBirth>10/05/1980</DateOfBirth>
  </Person>
  <Person PersonalID="37890">
    <Name>Sidor</Name>
    <Surname>Sidorov</Surname>
    <DateOfBirth>21/10/1978</DateOfBirth>
  </Person>
</Main>

```

Также возможно выгружать подчиненные таблицы, описав при этом связи родительской и дочерней таблиц. Это делается с помощью элемента *sql:relationship*, вложенного в элементы *xsd:annotation* и *xsd:appinfo*.

Тег *sql:relationship* должен содержать следующие атрибуты:

- *name* – имя отношения;

- parent – родительская таблица;
- parent-key – имя ключевого поля родительской таблицы;
- child – имя дочерней таблицы;
- child-key – имя поля дочерней таблицы, связанного с ключевым полем родительской таблицы.

Приведем пример:

```
<xs:schema xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sql="urn:schemas-microsoft-com:mapping-schema">
  <xs:annotation>
    <xs:appinfo>
      <sql:relationship name="LPU_TFOMS"
        parent="tmp_TFOMS"
        parent-key="TFOMSID"
        child="tmp_LPU"
        child-key="rf_TFOMSID" />

      <sql:relationship name="DOCTOR_LPU"
        parent="tmp_LPU"
        parent-key="LPUID"
        child="tmp_DOCTOR"
        child-key="rf_LPUID" />

    </xs:appinfo>
  </xs:annotation>

  <xs:element name="TFOMS_AI" sql:relation="tmp_TFOMS">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="LPU_AI" sql:relation="tmp_LPU"
          sql:relationship="LPU_TFOMS">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="DOCTOR_AI" sql:relation="tmp_DOCTOR"
                sql:relationship="DOCTOR_LPU">
                <xs:complexType>
                  <xs:attribute name="d_code" />
                  <xs:attribute name="d_name" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="l_ogrn"/>
            <xs:attribute name="l_name"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="t_ogrn" />
      <xs:attribute name="t_name" />
      <xs:attribute name="t_okato"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Б.4.9. Приемник DBF

Приемник DBF – это правило формирования DBF-файла и заполнение его данными из БД. Один приемник DBF может формировать только один DBF-файл.

В приемнике указывается путь и имя к файлу выгрузки, при этом можно использовать значения параметров, а также ряд других символов подстановки. По умолчанию файл выгружается в кодировке CP866, но при необходимости ее можно сменить на Windows-1251.

Для выборки данных в файл можно указать постоянную таблицу, временную таблицу или sql-запрос (Рисунок 127).

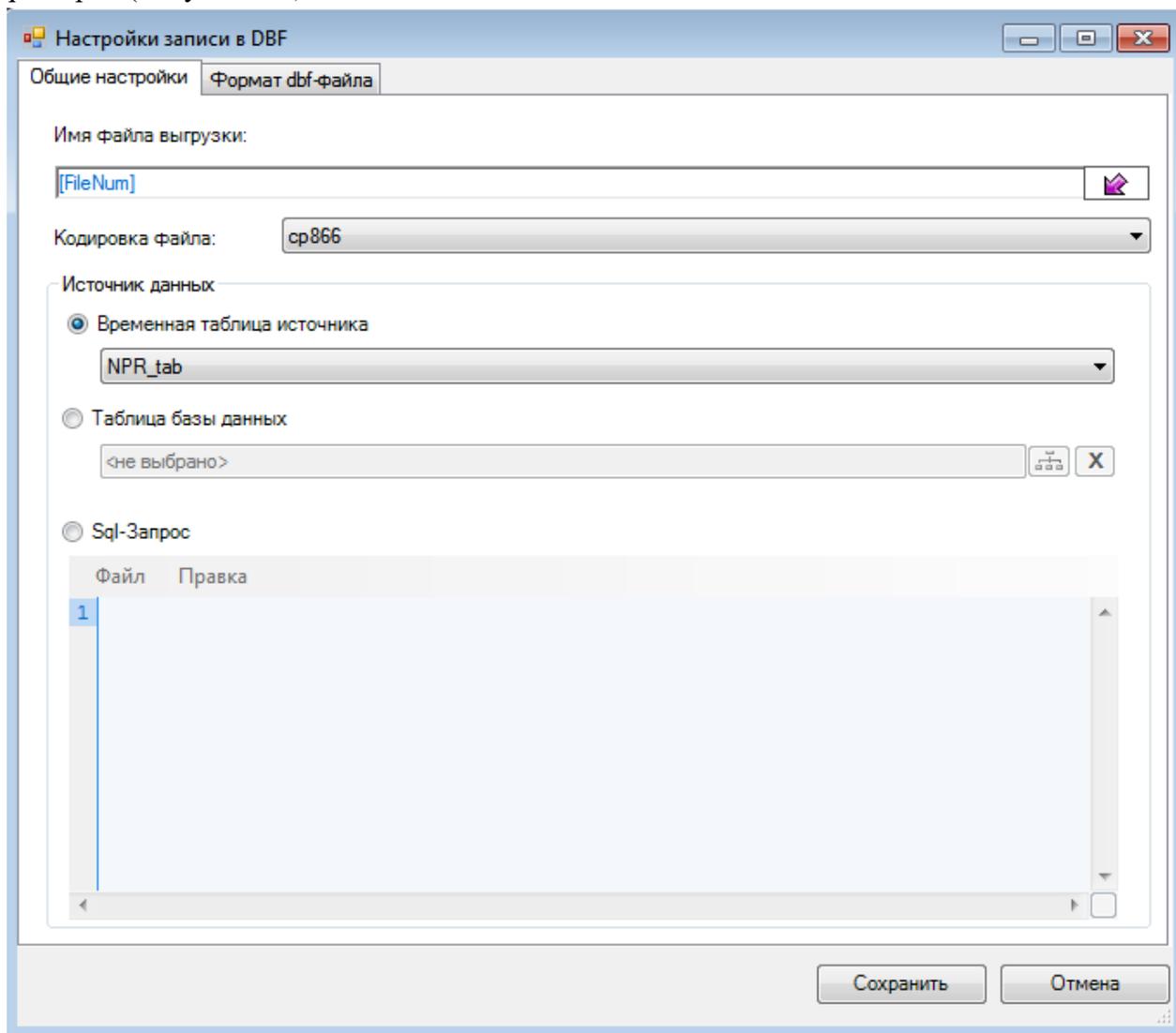


Рисунок 127. Диалог настройки выгрузчика DBF – вкладка «Общие настройки»

Структура получаемого dbf-файла может быть, как сгенерирована автоматически, так и задана пользователем (Рисунок 128). В последнем случае необходимо внимательно следить за тем, чтобы количество и порядок полей в выборке совпадали с количеством и порядком столбцов в структуре файла.

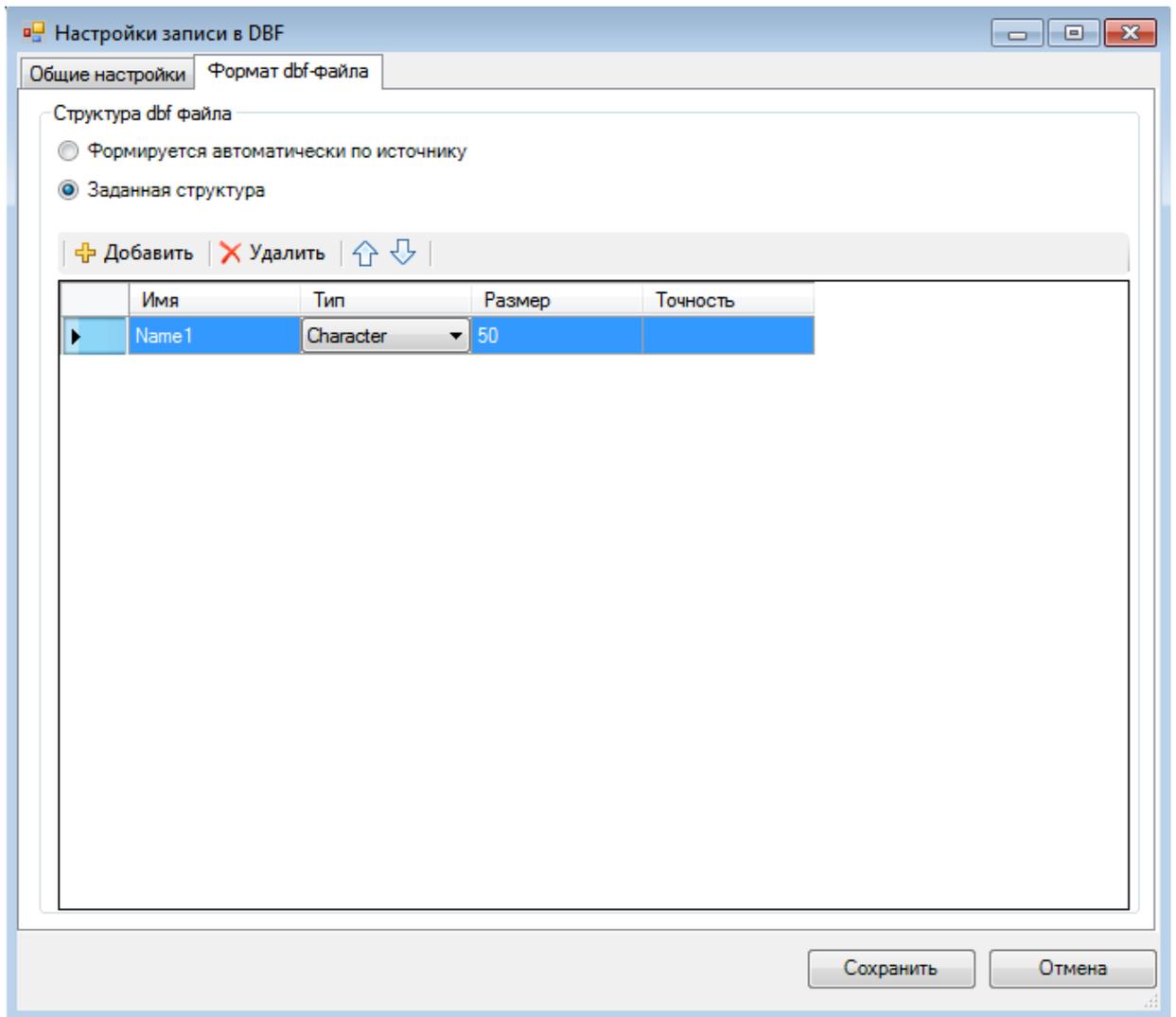


Рисунок 128. Диалог настройки выгрузчика DBF – вкладка «Формат dbf – файла»