

ΠЛΑΤΦΟΡΜΑ UNIDATA 3.5

РУКОВОДСТВО АДМИНИСТРАТОРА

ВЕРСИЯ 1.2 2016 год Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления.

Никакая часть данного документа не может быть воспроизведена или передана в любой форме и любыми способами в каких-либо целях без письменного разрешения ООО «ЮниДата».

© UniData, 2015 - 2016. Все права защищены.

оглавление

1	1 Общие положения			
1	.1	Краткое описание Платформы UniData6		
	1.1.1	Назначение, функции и возможности6		
	1.1.2	Состав и структура6		
1	.2	Администрирование ПО UniData7		
	1.2.1	Назначение и функции администратора7		
2	Оби	цая информация о работе системы8		
2	.1 3	Записи9		
	2.1.1	Общая структура9		
	2.1.2	Атрибуты и значения атрибутов9		
	2.1.3	Правила качества данных11		
	2.1.4	Версии16		
	2.1.5	Периоды актуальности17		
	2.1.6	Жизненный цикл		
	2.1.7	Связи21		
	2.1.8	Дубликаты22		
2	.2	Функции		
	2.2.1	Общая информация о Функциях22		
	2.2.2	Принципы работы с функциями23		
	2.2.3	Простые функции23		
	2.2.4	Сторонние функции23		
	2.2.5	Композитные функции24		
3	Адм	инистрирование системы		
3	.1)	/правление ролями		
	3.1.1	Создание роли25		
3	.2	/правление пользователями25		
	3.2.1	Создание пользователя25		
3	.3	Иандатный доступ		
4	Нас	гойка источников данных		
4	.1	Создание источников		
4	.2	Зеса источников		

5	Мод	ель данных	28
5	.1 4	1нтерфейс раздела схема данных	28
5	.2 (Создание справочника	29
	5.2.1	Настройка свойств справочника	29
	5.2.2	Настройка модели атрибутов справочника	30
	5.2.3	Настройка правил качества данных	35
	5.2.4	Настройка правил проверки на дубликаты	39
5	.3 (Создание реестра	39
	5.3.1	Настройка свойств реестра	39
	5.3.2	Настройка модели атрибутов	40
	5.3.3	Настройка связей	46
	5.3.4	Настройка правил качества данных	46
	5.3.5	Настройка правил проверки на дубликаты	50
6	Фун	кции	51
6	.1 ŀ	łастройка композитных функций	51
	6.1.1	Последовательность выполнения	51
6	.2 (Создание сторонних функций	51
	6.2.1	Последовательность выполнения	51
	6.2.2	Результат выполнения	52
	6.2.3	Устранение возможных нарушений	52
6	.3 F	едактирование сторонних функций	52
	6.3.1	Последовательность выполнения 1	52
	6.3.2	Последовательность выполнения 2	53
	6.3.3	Результат выполнения	53
	6.3.4	Устранение возможных нарушений	53
7	Рабо	ота с данными	54
7	.1 E	ведение	54
7	.2 F	аздел обработка данных	55
	7.2.1	Общая схема работы с данными	55
7	.3 Г	Тоиск записи	56
	7.3.1	Дополнительные условия при формировании поисковой вылачи	57
7	.4 (СRUD записи	57
	7.4.1	Создание записи	57

7.4.2 Просмотр записи
7.4.3 Редактирование записи
7.4.4 Удаление записи60
7.5 Работа с записями
7.5.1 Работа с ошибками качества данных61
7.5.2 Объединиие записей62
7.5.3 Работа с записями, являющимися потенциальными дубликатами
7.5.4 Просмотр исходных версий записи64
7.6 Версионирование записей
7.6.1 Создание новой версии записи с помощью интерфейса UniData (без периодов актуальности)
7.6.2 Создание новой версии записи с помощью интерфейса UniData (с созданием периодов актуальности)
7.6.3 Создание новой версии записи с помощью интерфейса UniData (с редактированием периодов актуальности)
Приложения67
Приложение А. Перечень терминов и сокращений67
Приложение Б. Простые функции

1 ОБЩИЕ ПОЛОЖЕНИЯ

1.1 КРАТКОЕ ОПИСАНИЕ ПЛАТФОРМЫ UNIDATA

1.1.1 НАЗНАЧЕНИЕ, ФУНКЦИИ И ВОЗМОЖНОСТИ

Платформа UniData предназначена для построения систем централизованного управления информацией.

К основным функциям предоставляемым платформой следует отнести:

• Функциональный блок «Администрирование»

В рамках данного блока платформа предоставляет инструментарий для управления пользователями и ролями.

• Функциональный блок «Управление данными»

Предоставляет функциональность для:

о Создания, просмотра и редактирования модели данных.

Модель данных определяет сущности НСИ, их структуру и взаимосвязи.

Также для каждой сущности задаются правила качества данных, привила поиска дубликатов и консолидации.

- о Управления источниками данных.
- Просмотр библиотеки функций очистки и обогащения данных.
- Функциональный блок «Обработка данных»

Поддержка функций поиска, создания, просмотра и редактирования записей.

• Интеграционный блок

Набор различных АРІ для интеграции с внешними системами. АРІ поддерживают все основные операции с данными.

1.1.2 СОСТАВ И СТРУКТУРА

Платформа UniData выполнена в виде клиент-серверного приложения с классической трехзвенной архитектурой и включает в себя:

• Клиентское приложение

Клиентское приложение предоставляет пользователям доступ ко всем основным функциям платформы:

- администрированию системы;
- управлению данными;
- обработке данных.

• Серверное приложение

Серверное приложение реализует основную логику платформы, предоставляет АРІ для доступа к данным системам-источникам и системам-получателям.

• Базу данных

База данных работает под управлением СУБД PostgreSQL.

• Поисковый индекс

Поисковый индекс развернут на базе ElasticSearch.

Клиентское приложение и Серверное приложение запускаются под управлением контейнера сервлетов Tomcat.

1.2 АДМИНИСТРИРОВАНИЕ ПО UNIDATA

1.2.1 НАЗНАЧЕНИЕ И ФУНКЦИИ АДМИНИСТРАТОРА

В задачи администратора платформы UniData входит:

- управление ролями;
- управление пользователями;
- управление расписанием запуска механизма поиска дубликатов;
- управление поисковый индексом.

2 ОБЩАЯ ИНФОРМАЦИЯ О РАБОТЕ СИСТЕМЫ

В системе UniData общая структура данных, трансформаций и взаимоотношений между объектами задается в модели данных. Настройки модели данных обуславливают правила работы с данными в системе. Основные объекты модели данных перечислены ниже:

РЕЕСТРЫ — записи с набором атрибутов и их значений, а также модель связей между записями. В модели реестра описываются привила валидации и обогащения данных.

СПРАВОЧНИКИ – набор атрибутов и их значений, основное отличие от реестров заключается в том, что к справочникам не применяются правила качества данных и нет связей.

ПЕРЕЧИСЛЕНИЯ – простой список возможных значений.

	Перечисление	Справочник	Реестр	Комментарий
Список элементов	+	+	+	Содержание нескольких элементов
Атрибуты	_	+	+	Содержание более одного атрибута, например, запись может хранить наименование компании, её ИНН, ОГРН, и прочие реквизиты
Правила качества данных	-	+	+	
Правила поиска дубликатов	-	+	+	
Ссылки	_	-	+	
Вложенные сущности	-	-	+	

СПИСОК ЭЛЕМЕНТОВ — перечень записей, принадлежащих одному объекту. Например, значения перечисления или записи в реестре.

АТРИБУТЫ – пара ключ-значение, типы и возможные значения атрибутов описаны ниже.

ПРАВИЛА КАЧЕСТВА ДАННЫХ — набор функций, связанный с атрибутами, и предназначенный либо для валидации данных, либо для их обогащения.

ПРОВЕРКА НА ДУБЛИКАТЫ — настройка правил проверки на наличие дубликатов, подразумевается полное соответствие по записям, по заданным в правилах атрибутам.

ССЫЛКИ – связи одной записи с другой записью в реестре.

ВЛОЖЕННЫЕ СУЩНОСТИ – набор сущностей, вложенных в одну запись. Например, сущность «Номер телефона» с атрибутами: тип, код города, номер – является вложенной к записи «Сотрудник». Сотрудник может иметь несколько телефонов: {домашний, +7495,6543219} и {мобильный, +7999,6543219}.

2.1 ЗАПИСИ

2.1.1 ОБЩАЯ СТРУКТУРА

Запись содержит информацию двух типов: системную информацию и данные.

СИСТЕМНАЯ ИНФОРМАЦИЯ

ID – внутрисистемный уникальный идентификатор записи.

ДАТА И ВРЕМЯ СОЗДАНИЯ – момент времени, когда запись появилась в системе UniData.

ДАТА ИЗМЕНЕНИЙ – момент времени, когда запись была изменена в системе UniData.

ПЕРИОДЫ АКТУАЛЬНОСТИ – промежуток времени, когда запись активна.

ВЕРСИИ – предыдущие версии записи.

ИСТОЧНИК ЗАПИСИ – информационная система, из которой была получена запись.

ДАННЫЕ

АТРИБУТЫ И ИХ ЗНАЧЕНИЯ – набор элементов типа ключ-значение.

ПРЕДЫДУЩИЕ ВЕРСИИ ЗАПИСИ – предыдущие наборы данных.

ПЕРИОДЫ АКТУАЛЬНОСТИ ВЕРСИИ – период времени, когда та или иная версия записи активна.

2.1.2 АТРИБУТЫ И ЗНАЧЕНИЯ АТРИБУТОВ

Атрибуты задаются при создании модели данных и определяют какие данные могут содержаться в записи. Каждый атрибут обладает свойствами и типом.

ИМЯ — системный идентификатор атрибута. Используется в системе, а также используется в API.

Имя	
Отображаемое	
Описание	
Обязательный	да
Поисковое	нет
Отображаемое	нет
Гл.отображае	нет
Тип атрибута	

ОТОБРАЖАЕМОЕ ИМЯ — человеко-понятный идентификатор атрибута, который отображается в интерфейсе UniData. Это имя, которое пользователь будет видеть в качестве названия атрибута.

ОПИСАНИЕ – дополнительная информация о данном атрибуте.

ПОИСКОВОЕ – настройка определяющая возможность поиска по данному атрибуту. Поиск может осуществляться как с помощью основного поискового поля, так и с применением поатрибутного поиска. В том случае, когда данный параметр выбран активным, при просмотре раздела данных этот параметр автоматически появится в панели указания значений для поатрибутного поиска.



ОТОБРАЖАЕМОЕ — атрибут, который будет отображаться в поисковой выдаче, и позволит пользователю идентифицировать объект.

ГЛАВНОЕ ОТОБРАЖАЕМОЕ — основной идентифицирующий атрибут, который будет отображаться в поисковой выдаче.

Следующей частью создания атрибута, является указание его типа.

ПРОСТОЙ АТРИБУТ – имя этого атрибута является ключом, ниже перечислены типы возможных значений:

СТРОКОВОЕ – строка без дополнительных ограничений, длина 2048 символов.

ЦЕЛОЧИСЛЕННОЕ – целочисленное значение, длина 2⁶⁴.

ЧИСЛЕННОЕ – числовое значение, длина 2⁶⁴.

ЛОГИЧЕСКОЕ – может принимать следующие значение: true, false, null.

ДАТА/ВРЕМЯ – дата в формате ISO timestamp.

ДАТА – дата.

ВРЕМЯ – время.

БИНАРНЫЙ МАССИВ — это приложенный файл, поиск по которому не выполняется.

ТЕКСТОВЫЙ МАССИВ – массив текстовых данных, хранящийся в базе данных.

ССЫЛКА НА СПРАВОЧНИК – указывается конкретный справочник, в котором будет осуществляться поиск значения атрибута.

ПЕРЕЧИСЛЕНИЕ – простой список возможных значений атрибута.

ССЫЛКА НА ВЕБ-РЕСУРС – шаблон ссылки на сторонний ресурс (адрес и переменная), в записи из шаблона формируется ссылка с использованием данных записи.

ВЛОЖЕННЫЕ СУЩНОСТИ (КОМПЛЕКСНЫЙ АТРИБУТ) — набор сущностей, вложенных в одну запись. Например, сущность «Номер телефона» с атрибутами: тип, код города, номер является вложенной к записи «Сотрудник». Сотрудник может иметь несколько телефонов: {домашний, +7495,6543219} и {мобильный, +7999,6543219}.

2.1.3 ПРАВИЛА КАЧЕСТВА ДАННЫХ

Правила качества данных сопоставляют конкретные атрибуты записи и функции для их стандартизации или валидации. Например, в данных есть идентификатор с контрольной суммой и без. Стандартизировать их можно, очищая от контрольной суммы.

— Свойства правила	
Имя правила:	whole area roudup
Описание:	
-	

ИМЯ – отображаемое название правила качества данных.

ОПИСАНИЕ – текстовое описание выполняемых действия со значениями атрибутов.

Настройки правила			
Функция:	Разное.ПроверкаИНН	~	
🖂 Валидация	🖂 Обогащение		

ФУНКЦИЯ – выполняемая этим правилом функция, подробнее о функциях можно прочитать в разделе Функции.

РЕЖИМ РАБОТЫ — валидация - это выполняемая проверка значения; обогащение - это трансформация значения. Большинство функций может работать только в одном значении.

Источники	
☑ Bce	
системы	
EGAIS	
🗌 unidata	

ИСТОЧНИКИ – при необходимости можно применять правило качества данных для конкретных систем. Например, это может потребоваться для стандартизации данных, поступающих из разных систем.

Атрибуты

— 🗁 Технико-экономические паспорта многоквартирных домов			
Улица (string, обязательный)			
Дом (string, обязательный)			
— Корпус (integer)			
Литера (string)			
Район (спр.: "districts-spb", обязательный)			
Серия, тип проекта (спр.: "house-project-types")			
Год постройки (integer, обязательный)			
Год проведения реконструкции (integer)			
Общая площадь здания (number, обязательный)			
:			

Входящие порты

NHH*:			
порт	: string	\otimes	
конст	танта	\otimes	

Исходящие порты

Результат выполнения функции:	
порт: boolean	\otimes

АТРИБУТЫ — выбор атрибутов, к значениям которых будет применяться выбранная функция, а также выбор атрибутов, в значения которых будет выводиться результат выполнения функции. Стоит отметить, что валидационные функции имеют выходной логический порт, в который передается результат проверки. Его не обязательно сопоставлять с атрибутом в записи.

• ВАЛИДАЦИЯ

Валидация	
Ошибка	Основные
Создавать ошибку на основе:	Текст сообщения:
port1 V	- порт не выбран
Фаза обработки данных:	Ошибка ИНН
до сохранения исходных данных 🗸	Критичность:
	высокая
	Категория:
	- порт не выбран -
	Базовые ошибки
0.000	

Ошиока	
Создавать ошибку на основе:	
port1	~

ВАЛИДАЦИЯ — этот блок появляется при выборе режима «Валидация». Необходимо выбрать на основании чего (какого порта) создается ошибка, для этого выбрать логический порт. Если в порт приходит значение false, то валидационное правило выдает ошибку.

Фаза обработки данных:

до сохранения исходных данных

ФАЗА ОБРАБОТКИ ДАННЫХ - если фаза выбрана до сохранения исходных данных, то при попытке сохранения записи, запись с ошибкой в данном правиле сохранена не будет. Если выбрана фаза после сохранения, то запись с ошибкой будет сохранена, и будет выставлен "флаг" ошибочной записи. Если выбрана фаза после консолидации, то проверка будет выполняться только после консолидации, то есть запись с ошибкой будет сохранена, и будет выставлен "флаг" ошибочной записи.

Текст сообщения:

- порт не выбран -

Ошибка ИНН

ТЕКСТ СООБЩЕНИЯ - значение порта функции, или текст, указанный администратором. Большинство простых функций не могут самостоятельно выдать текст об ошибке, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать текст об ошибке, это необходимо, так как возможно различные ошибки и тексты сообщения должны отличаться в разных случаях.

Ú

Критичность:

высокая

КРИТИЧНОСТЬ ОШИБКИ - значение порта функции, или критичность, указанная администратором. Большинство простых функций не могут самостоятельно выдать критичность ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать критичность ошибки, это необходимо, так как возможны различные ошибки, когда критичность ошибки должна отличаться при разных ошибках.

Категория:

КАТЕГОРИЯ ОШИБКИ - значение порта функции, или категория, указанная администратором. Большинство простых функций не могут самостоятельно определить категорию ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут передавать категорию ошибки, это необходимо, так как возможны различные ошибки и категории ошибок будут разные.

• ОБОГАЩЕНИЕ

\cap	50	ro.			LI I	LA.	
	50	a	щ	C	п	n	С

CEGAIS		
🔾 unidata		
При обогащении фиксировать изменения:		
○ Изменять имеющуюся сущность		
○ Создавать новую сущность		
Фаза обработки данных:		
- выберите фазу -		

ОБОГАЩЕНИЕ – этот блок появляется, при выборе режима «Обогащение».



СИСТЕМА ИСТОЧНИК – система, от имени которой публикуются изменения. Это влияет не только на наименование, но и на то какой вес будет иметь измененное значение атрибута.

При обогащении фиксировать изменения:

О Изменять имеющуюся сущность

О Создавать новую сущность

ФИКСАЦИЯ ИЗМЕНЕНИЙ - это может быть, как изменение имеющейся сущности, так и создание новой.

Фаза обработки данных:

до сохранения исходных данных

ФАЗА ОБРАБОТКИ ДАННЫХ – в режиме «Обогащения» правило может быть применено только в режиме «До сохранения исходных данных». Данный режим означает, что изменения будут сохранены вместе с записью.



Версия истины – это некий набор данных и служебных данных, описывающих некоторое истинное состояние объекта в определенный промежуток времени. Например, водительское удостоверение является версией, так как оно характеризуется следующим набором данных натуральный идентификатор (номер), дата создания (дата выдачи), период действия (10 лет с момента выдачи), источник данных (орган, выдавший в/у), данные (ФИО, категории и прочее).

Источник истины – это набор данных, собранный из версий истин, по определенным правилам. В системе существует понимание жизненного цикла, и в связи с этим, источник отдает ту или иную версию истины, в зависимости от даты и времени, указанных в запросе. *Например, вы получили водительское удостоверение (2008-2018), а потом вы его потеряли (сент. 2012) и получили новое (окт. 2012). Итого у вас три версии истины сначала: в/у со*

сроком действия 10 лет, потом в/у присвоили статус недействительного, и потом вы получили новое удостоверение. Если мы сформируем запрос указанием 2010 года, то вернется информация по первому удостоверению, если без указания, то система автоматически подставит текущую дату и время, и вернется информация по новому удостоверению.

Эталон истины – это набор данных собранный из нескольких источников, по определенным правилам. В нашей системе кроме времени, в расчетах истины принимает участие также вес источника. Например, водительское удостоверение изъято решением суда, соответственно появляется запись о том, что в/у с таким номером изъято, так как система публикации судебный решений имеет больший вес, то на время лишения, эталоном будет считаться отсутствие права управления TC.

2.1.5 ПЕРИОДЫ АКТУАЛЬНОСТИ

Период актуальности определяет актуальность версии на определенном временном промежутке.



Например, вы получили водительское удостоверение (2008-2018) эта версия имеет период актуальности с 2008 по 2018,



а потом вы его потеряли (сент. 2012), мы создали новую версию записи о в/у, которая делает предыдущую версию не действительной с сент 2012. В итоге в/у с тем номером считается недействительным.



и получили новое (окт. 2012). С новым номером, которое действительно до 2022 года.

Итого у вас три версии истины сначала: в/у со сроком действия 10 лет, потом в/у присвоили статус недействительного, и потом вы получили новое удостоверение. Если мы сформируем запрос указанием 2010 года, то вернется информация по первому удостоверению, если без указания, то система автоматически подставит текущую дату и время, и вернется информация по новому удостоверению.

2.1.6 ЖИЗНЕННЫЙ ЦИКЛ

ОПИСАНИЕ ЖИЗНЕННОГО ЦИКЛА ИСТОЧНИКА

Жизненный цикл сущности в системе UniData отражает сложность жизненного цикла объектов в реальном мире. В связи с этим у сущности есть версии, и в процессе жизненного цикла, у сущности могут меняться параметры, что отражается в системе. Каждая версия сущности может иметь свои периоды актуальности, которые сводятся в единый жизненный цикл.

ИСТОРИЯ ИЗМЕНЕНИЙ И ЖИЗНЕННЫЙ ЦИКЛ

Изменения в версиях одновременно формируют как жизненный цикл, так и историю изменений.





🛠 ПРАВИЛА РАБОТЫ ЖИЗНЕННОГО ЦИКЛА ИСТОЧНИКА

ПРАВИЛО 1: ПРЕИМУЩЕСТВО ПОСЛЕДНЕЙ ВЕРСИИ



19

В рамках одного источника преимущество всегда у последней версии истины. Поэтому в зависимости от времени, указанного в запросе, данные могут меняться.

ПРАВИЛО 2: ВРЕМЕННОЕ НАПРАВЛЕНИЕ НЕ ВЛИЯЕТ



Вне зависимости от того, на какой период была добавлена последняя версия истины, она имеет преимущественное значение на этот период.

ПРАВИЛО 3: ПЕРЕКРЫТИЕ НЕ ВЛИЯЕТ НА ВЕРСИЮ



После появления новой версии с меньшим периодом действия, предыдущая версия остается действительной на не перекрытых временных отрезках.

ПРАВИЛО 4: ПЕРЕКРЫТЫЕ ВЕРСИИ НЕ ВЛИЯЮТ НА ПЕРИОДЫ



В расчетах конечной версии источника участвуют только последние версии истины на каждом из временных промежутков. Также количество временных промежутков определяется количеством итоговых версий истины и временных промежутков этих версий. В данном примере в источнике есть два периода и две версии.

2.1.7 СВЯЗИ

Связи — это отсылка к другому объекту. Связи бывают двух типов: обычная и связь типа включение. Например, связь между юридическим лицом и отделом ФНС.

Типы записей и целевые реестры задаются в процессе моделирования. Для установления связи между объектами необходимо либо выбрать объект вручную, либо при импорте данных передавать уникальный идентификатор для записи, с которой устанавливается связь.



ОБЫЧНАЯ СВЯЗЬ связывает две записи. Такая связь кроме обычных параметров имеет свои собственные периоды актуальности и версии. Период актуальности связи не связан с периодами актуальности записей, и может быть, как больше, так и меньше их. Связи устанавливаются между записями реестров. Запись может быть связана как с записью этого реестра, так и с записью другого реестра.



СВЯЗЬ ТИПА ВКЛЮЧЕНИЕ устанавливает связь между основным объектом и подчиненным объектом. На данную связь накладываются ограничения на периоды актуальности. Они не должны превышать периоды актуальности основного объекта. Также накладывается ограничение на подчиненный объект, его период актуальности не может превышать самый поздний период актуальности основной записи. Отдельно стоит отметить, что в отличии от обычной связи, связь типа включение, не может связывать объекты из одного реестра.

2.1.8 ДУБЛИКАТЫ

В системе UniData можно проверять до пяти атрибутов на полное совпадение значений атрибутов записей одного реестра.

Имя	Описание
Общие	
Основные	
Имя правила:	
Описание:	
Атрибуты	Вхолящие порты
, appropriate	Diognation repres
🗄 🗁 Технико-экономические паспорта многоквартирных домов	Входное значение 1:
— У лица (string, обязательный)	\otimes
— Дом (string, обязательный)	Входное значение 2:
Kopnyc (integer)	\otimes
Литера (string)	Входное значение 3:
— Район (спр.: "districts-spb", обязательный)	8
Серия, тип проекта (спр.: "house-project-types")	Входное значение 4:
— Год построики (integer, ооязательный)	Входное значение 5:
– Год проведения реконструкции (integer)	\otimes
Общая площадь здания (number, обязательный)	
— Площадь жилых помещений (number)	

2.2 ФУНКЦИИ

2.2.1 ОБЩАЯ ИНФОРМАЦИЯ О ФУНКЦИЯХ

В системе UniData, функции применяются к значениям атрибутов в записях, для их стандартизации или валидации в правилах качества данных.

Типы функций:

ПРОСТЫЕ ФУНКЦИИ — стандартные функции системы UniData. Простые функции могут обрабатывать данные или проверить их, например, можно перевести текстовое поле в верхний регистр.

СТОРОННИЕ ФУНКЦИИ - представляют из себя исполняемый JAR файл, загруженный в систему.

КОМПОЗИТНЫЕ ФУНКЦИИ - композитные функции представляют собой последовательность исполняемых функций. Могут состоять из всех типов функций.

Для удобства работы, все функции организованы в группы.

2.2.2 ПРИНЦИПЫ РАБОТЫ С ФУНКЦИЯМИ

Все функции имеют один и более входной порт и один или более выходной порт.

Во входной порт функции передается значение атрибута, в выходной порт передаётся результат работы функции. Все функции по своему назначению делятся на две группы: валидационные (проверяющие значение) и обогащающие (изменяющие значение атрибута), а некоторые функции могут работать в обоих режимах.

В композитной функции результаты работы одной функции могут передаваться во входные порты следующей функции.

Выходными портами функции могут быть не только данные, но и результаты валидации. Для передачи результатов валидации в системе предназначен логический порт.

2.2.3 ПРОСТЫЕ ФУНКЦИИ

Детальный список простых функций представлен в приложении Б.

2.2.4 СТОРОННИЕ ФУНКЦИИ

Сторонняя функция – это загруженный в систему JAR файл. (Информация о том, как именно должен быть создан файл, содержится в руководстве по инсталляции)

После загрузки файла в систему, из него автоматически извлекаются название и описание функции, а также информация о портах функции.

После корректного распознавания файла, администратор может применять стороннюю функцию, так же, как и любую другую функцию.

Обновление сторонней функции выполняется путем загрузки нового файла с тем же JAVA методом.

2.2.5 КОМПОЗИТНЫЕ ФУНКЦИИ

Композитные функции – это последовательности исполняемых функций, объединенные в одну для удобства последующего применения.



Пример функции, которая соединяет 4 атрибута адреса в единый строковый атрибут.

Слева находятся входные порты функции, в центре - функции, которые составляют композитную функцию, а справа - выходной порт функции.

3 АДМИНИСТРИРОВАНИЕ СИСТЕМЫ

В системе UniData применяется ролевая модель безопасности.

ПОЛЬЗОВАТЕЛИ – пользователь получает права в системе UniData, согласно имеющимся у него ролям. Пользователь обладает уникальным именем в системе – login. Пользователь может обладать более чем одной ролью в системе.

РОЛИ – роль определяет объекты доступа, такие как администрирование системы, администрирование данных, а также права к реестрам.

ПРАВА – право на просмотр объекта, право на создание, на редактирование и право на удаление объекта. Так же есть полные права, являющиеся совокупностью четырёх предыдущих.

3.1 УПРАВЛЕНИЕ РОЛЯМИ

3.1.1 СОЗДАНИЕ РОЛИ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для создания роли необходимо:

- 1. Зайти в раздел управление ролями.
- 2. Нажать кнопку добавить новую роль.
- 3. Ввести логическое имя роли.
- 4. Ввести отображаемое имя роли. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человекоидентифицируемое название, например, администратор или оператор данных.
- 5. Проставить необходимые права.
- 6. Сохранить.

3.2 УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ

3.2.1 СОЗДАНИЕ ПОЛЬЗОВАТЕЛЯ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для создания пользователя необходимо:

- 1. Зайти в раздел управление пользователями.
- 2. Нажать кнопку добавить нового пользователя.
- 3. Указать ему статус активности.
- 4. Ввести имя и фамилию пользователя.
- 5. Ввести уникальный идентификатор login.

- 6. Ввести e-mail пользователя.
- 7. Ввести пароль и его подтверждение. Внимание: пароль и его подтверждение должны совпадать.
- 8. Указать роли, которыми обладает пользователь.
- 9. Сохранить.

3.3 МАНДАТНЫЙ ДОСТУП

Мандатный доступ позволяет ограничивать пользователя при работе с объектами. Для начала настройки мандатного доступа необходимо проставить метки безопасности.

МЕТКИ БЕЗОПАСНОСТИ – устанавливаются на атрибуты при импорте модели данных. Информация об установке меток содержится в руководстве по инсталляции.

После установки меток безопасности на реестры, необходимо в разделе управление ролями настроить использование меток для данной роли. Выбрать и применить необходимые метки безопасности.

После это необходимо у каждого пользователя указать допустимые значения меток.

На этом настройка меток безопасности заканчивается

4 НАСТОЙКА ИСТОЧНИКОВ ДАННЫХ

Источник данных – наименование системы, которая содержат исходные данные, и от имени которой они загружаются в систему UniData. Данные могут быть импортированы в систему с помощью API или путём загрузки файла с данными в систему. Система UniData собирает статистки по работе с источниками и отслеживает количество импортированных записей, а также разделяет новые и обновленные записи.

При создании источника необходимо указать его вес, и в зависимости от качества, доверенности и актуальности необходимо указать его вес. При автоматической консолидации дубликатов, эталоном будет становиться запись из источника с большим весом. Стоит отметить, что система UniData имеет вес 100, что позволяет перезаписывать данные пришедшие из других систем, у которых значения веса находятся в диапазоне от 0 до 99.

4.1 СОЗДАНИЕ ИСТОЧНИКОВ

Для начала импорта данных необходимо создать источники, от имени которых будут публиковаться записи.

Для создания источника необходимо:

- 1. Выполнить вход в раздел источники.
- 2. Нажать кнопку добавить источник.
- 3. Ввести его имя и описание.
- 4. Указать вес источника. В зависимости от качества и доверенности источника администратор указывает вес источника от 0 до 99. При автоматической консолидации дубликатов, эталоном будет становиться запись из источника с большим весом.
- 5. На этом работа с источником заканчивается, необходимо сохранить источник.

4.2 ВЕСА ИСТОЧНИКОВ

Веса источников влияют на автоматическое объединение дубликатов, при котором версии полученные из источников с большим весом станут эталонами.

Источники данных	3 🕀	unidata × EGAIS	S ×		
EGAIS					
🖭 unidata		Имя:	EGAIS	din .	
		Описание:	<u>ЕГАИС</u> здравоохранен	ния	
		Bec:	55	Carlo I	

5 МОДЕЛЬ ДАННЫХ

5.1 ИНТЕРФЕЙС РАЗДЕЛА СХЕМА ДАННЫХ

🛠 ОБЩИЙ ВИД

	Выход	
		TE-pasports-for-block-of-flats
	Обработка данных	Технико-экономические паспорта многоквартирных домов
¢	Главная	Идентификатор ЕАС : ЕАС-Id
		Дата составления ТЭП : TE-pasport-creation-date
≣	Данные	Аварийность дома : house-emergency
-		Год модернизации лифта : elevator-modernosation-year
*	Импорт / Экспорт	Год реконструкции лифта : elevator-renovation-year
	Управление данными	
ń.	Модель данных	
*	Схема данных	
_		
N	Источники данных	
	*	
Q	Функции	
		ст-турез отвілістів-вро і проектов Районы г. Санкт-Петербурга
	Администрирование	
*	Пользователи	
a,	Роли	

СПИСОК РЕЕСТРОВ И СПРАВОЧНИКОВ



5.2 СОЗДАНИЕ СПРАВОЧНИКА

5.2.1 НАСТРОЙКА СВОЙСТВ СПРАВОЧНИКА

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для создания справочника и настройки его свойств необходимо:

1. Выполнить вход в раздел модель данных.



- 2. Нажать кнопку добавить справочник/реестр.
- 3. Выбрать, сущность какого типа необходимо создать: справочник.
- 4. Пользователь начинает работу во вкладке свойства.
- 5. Задать системное имя справочника, допускается использовать только латиницу и цифры.
- 6. Задать отображаемое имя справочника, это то имя справочника, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Рекомендуется использовать человеко-идентифицируемое название, например, контрагенты или сотрудники.
- 7. Задать описание этого справочника, которое используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени справочника недостаточно

понятно, какие объекты содержатся в данном справочнике, то рекомендуется заполнить описание справочника.

- 8. Отображение статистки для оператора данных. Дополнительная настройка, которая не влияет на метамодель. Если Отображение статистики включено, то на главной странице обработки данных появится блок статистики по данному справочнику, где будет указано количество записей, ошибок и потенциальных дубликатов.
- 9. Сохранить.

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

Результатом выполнения данной операции является справочник готовый к моделированию.

5.2.2 НАСТРОЙКА МОДЕЛИ АТРИБУТОВ СПРАВОЧНИКА

НАСТРОЙКА КОДОВОГО АТРИБУТА

Для настройки кодового атрибута справочника выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит к вкладке атрибуты.
- 4. Нажимает кнопку создать кодовый атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.
- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- 10. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному справочнику появится поле(-я) для поиска по данному атрибуту.

- 11. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 12. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятный идентификатор, например, название компании, а не её ИНН.
- 13. Указывается тип значения кодового атрибута. Значение может быть текст, целочисленное, численное значение.
- 14. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

НАСТРОЙКА ПРОСТОГО АТРИБУТА

Для настройки простого атрибута справочника выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит к вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.

Имя	
Отображаемое	
Описание	
Обязательный	нет
Только для чте…	нет
Скрытый	нет
Поисковое	нет
Отображаемое	нет
Гл.отображае	нет
Тип атрибута	

- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.

- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.
- 10. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.
- 11. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- 12. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному справочнику появится поле(-я) для поиска по данному атрибуту.
- 13. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 14. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятный идентификатор, например, название компании, а не её ИНН.
- 15. Задается типа атрибута простой.
- 16. Указывается тип значения простого атрибута. Значение может быть текст, дата, целочисленное, численное значение, логическое значение, бинарные данные, большое текстовое значение.
- 17. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

НАСТРОЙКА АТРИБУТА ТИПА ПЕРЕЧИСЛЕНИЕ

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит ко вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а

также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.

- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.
- 10. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.
- 11. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- 12. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному справочнику появится поле(-я) для поиска по данному атрибуту.
- 13. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 14. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятные идентификатор, например, название компании, а не её ИНН.
- 15. Задается типа атрибута перечисление. Само перечисление нужно создать до этого момента.
- 16. Выбирается перечисление из списка перечислений.
- 17. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

НАСТРОЙКА АТРИБУТА ТИПА ССЫЛКА НА ВЕБ-РЕСУРС

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит ко вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а

также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.

- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.
- 10. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.
- 11. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- 12. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному справочнику появится поле(-я) для поиска по данному атрибуту.
- 13. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 14. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятные идентификатор, например, название компании, а не её ИНН.
- 15. Задается тип атрибута ссылка на веб-ресурс.
- 16. Вводится URL в виде <u>http://www.example.ru/</u>
- 17. Вводится имя атрибута, который будет использоваться в качестве переменной для указания конкретной сущности на веб-ресурсе, в виде {attr_name}.
- 18. Итоговый результат шаблон вида <u>http://www.example.ru/{attr_name}</u>
- 19. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

НАСТРОЙКА АТРИБУТА ТИПА ССЫЛКА НА СПРАВОЧНИК

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит ко вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.

- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.
- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.
- 10. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.
- 11. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- 12. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному справочнику появится поле(-я) для поиска по данному атрибуту.
- 13. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 14. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятные идентификатор, например, название компании, а не её ИНН.
- 15. Задается типа атрибута ссылка на справочник. Внимание справочник должен быть уже создан.
- 16. Выбирается справочник из списка справочников. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

5.2.3 НАСТРОЙКА ПРАВИЛ КАЧЕСТВА ДАННЫХ

НАСТРОЙКА ПРАВИЛ ВАЛИДАЦИИ

Для настройки правил валидации выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке качество данных.

- 4. Нажимаем кнопку добавить правило.
- 5. Вводит имя правила допускается использовать только латиницу и цифры. Рекомендуется использовать человеко-понятное наименование правила.
- 6. Вводит описание правила. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из имени правила и выбранной функции недостаточно для понимания, того какое действие выполняет данное правило, то рекомендуется заполнить описание правила, например, правило "проверка ИНН" описание должно содержать следующую информацию "проверяется контрольная сумма в значении ИНН", это важно так как ИНН можно проверить и другим способом.
- 7. Выбираем выполняемую функцию. Перед созданием правил качества данных рекомендуется ознакомиться с возможностями функций.
- 8. Выбирается режим, в котором будет работать правило валидация, обогащение или валидация и обогащение. В нашем случае мы выбираем валидацию. Будьте внимательны большинство функции работают только в одном режиме и выбор режима несоответствующего функции делает правило лишенным смысла. Например, функция проверки контрольной суммы ИНН не может работать в режиме обогащения.
- 9. Выбираются источники данных, к которым применяет данное правило качества данных. Так как в один в один справочник нам могут поступать данные из разных источников, то к ним могут применяться разные правила качества данных, например, дата в одном источника хранится как MM/DD/YYYY, а в другом DD.MM.YYYY, для корректной обработки дат из разных систем мы должны привести их к одному формату.
- 10. Следующий этап это маппинг (сопоставление) атрибутов справочника и портов функции.
- 11. Необходимо выбрать на основании какого порта создается ошибка. Необходимо выбрать логический порт. В том случае если в порт приходит значение false то, это значит, что валидационное правило выдало ошибку.
- 12. Выбирается фаза обработки данных. Если выбрана фаза до сохранения исходных данных, то при попытке сохранения записи, запись с ошибкой в данном правиле сохранена не будет. Если выбрана фаза после сохранения, то запись с ошибкой будет сохранена, и будет выставлен "флаг" ошибочной записи.
- 13. Выбирается текст сообщения, есть возможность транслировать значение порта функции, или ввести текст самостоятельно. Большинство простых функций не могут самостоятельно выдать текст об ошибке, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать текст об ошибке, это необходимо, так как возможны различные ошибки и тексты сообщения должны отличаться в разных случаях.
- 14. Выбирается критичность ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно выдать критичность ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать

критичность ошибки, это необходимо, так как возможны различные ошибки и критичность ошибки должна отличаться при разных ошибках.

- 15. Выбирается категория ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно определить категорию ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут передавать категорию ошибки, это необходимо, так как возможны различные ошибки и категории ошибок будут разные.
- 16. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

НАСТРОЙКА ПРАВИЛ ОБОГАЩЕНИЯ

Для настройки правил обогащения выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке качество данных.
- 4. Нажимаем кнопку добавить правило.
- 5. Вводит имя правила допускается использовать только латиницу и цифры. Рекомендуется использовать человеко-понятное наименование правила.
- 6. Вводит описание правила. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из имени правила и выбранной функции недостаточно для понимания, того какое действие выполняет данное правило, то рекомендуется заполнить описание правила, например, правило "проверка ИНН" описание должно содержать следующую информацию "проверяется контрольная сумма в значении ИНН", это важно так как ИНН можно проверить и другим способом.
- 7. Выбираем выполняемую функцию. Перед созданием правил качества данных рекомендуется ознакомиться с возможностями функций.
- 8. Выбирается режим, в котором будет работать правило валидация, обогащение или валидация и обогащение. В нашем случае мы выбираем обогащение. Будьте внимательны большинство функции работают только в одном режиме и выбор режима несоответствующего функции делает правило лишенным смысла. Например, функция проверки контрольной суммы ИНН не может работать в режиме обогащения.
- 9. Выбираются источники данных, к которым применяет данное правило качества данных. Так как в один в один справочник нам могут поступать данные из разных источников, то к ним могут применяться разные правила качества данных, например, дата в одном источника хранится как MM/DD/YYYY, а в другом DD.MM.YYYY, для корректной обработки дат из разных систем мы должны привести их к одному формату.
- 10. Следующий этап это маппинг (сопоставление) атрибутов справочника и портов функции.
- 11. Выбирается фаза обработки данных.
- 12. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

НАСТРОЙКА ПРАВИЛ ВАЛИДАЦИИ И ОБОГАЩЕНИЯ

Для настройки правил валидации и обогащения выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке качество данных.
- 4. Нажимаем кнопку добавить правило.
- 5. Вводит имя правила допускается использовать только латиницу и цифры. Рекомендуется использовать человеко-понятное наименование правила.
- 6. Вводит описание правила. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из имени правила и выбранной функции недостаточно для понимания, того какое действие выполняет данное правило, то рекомендуется заполнить описание правила, например, правило "проверка ИНН" описание должно содержать следующую информацию "проверяется контрольная сумма в значении ИНН", это важно так как ИНН можно проверить и другим способом.
- 7. Выбираем выполняемую функцию. Перед созданием правил качества данных рекомендуется ознакомиться с возможностями функций.
- 8. Выбирается режим, в котором будет работать правило валидация, обогащение или валидация и обогащение. В нашем случае мы выбираем валидацию и обогащение. Данный выбор означает, что функция может работать в двух режимах, и в зависимости от данных. Если в данных содержится ошибка, то отработает только валидационная часть функции, если ошибки нет, отработает валидационная и обогащающая часть функции.
- 9. Выбираются источники данных, к которым применяет данное правило качества данных. Так как в один в один справочник нам могут поступать данные из разных источников, то к ним могут применяться разные правила качества данных, например, дата в одном источника хранится как MM/DD/YYYY, а в другом DD.MM.YYYY, для корректной обработки дат из разных систем мы должны привести их к одному формату.
- 10. Следующий этап это маппинг (сопоставление) атрибутов справочника и портов функции.
- 11. Необходимо выбрать на основании какого порта чего создается ошибка. Необходимо выбрать логический порт. В том случае если в порт приходит значение false то, это значит, что валидационное правило выдало ошибку.
- 12. Выбирается фаза выполнения валидационной составляющей функции. Если выбрана фаза до сохранения исходных данных, то при попытке сохранения записи, запись с ошибкой в данном правиле сохранена не будет. Если выбрана фаза после сохранения, то запись с ошибкой будет сохранена, и будет выставлен "флаг" ошибочной записи.
- 13. Выбирается текст сообщения, есть возможность транслировать значение порта функции, или ввести текст самостоятельно. Большинство простых функций не могут самостоятельно выдать текст об ошибке, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать текст об ошибке, это необходимо, так как возможны различные ошибки и тексты сообщения должны отличаться в разных случаях.

- 14. Выбирается критичность ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно выдать критичность ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать критичность ошибки, это необходимо, так как возможны различные ошибки и критичность ошибки должна отличаться при разных ошибках.
- 15. Выбирается категория ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно определить категорию ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут передавать категорию ошибки, это необходимо, так как возможны различные ошибки и категории ошибок будут разные.
- 16. Выбирается фаза выполнения обогащающей составляющей функции.
- 17. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

5.2.4 НАСТРОЙКА ПРАВИЛ ПРОВЕРКИ НА ДУБЛИКАТЫ

• ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для настройки правил проверки на дубликаты выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке дубликаты.
- 4. Выбирает ручное или автоматическое правило поиска дубликатов
- 5. Следующий этап это маппинг (сопоставление) атрибутов справочника и портов функции. Выбор тех атрибутов, которые будут проверяться на совпадение.
- 6. Если на этом работа с справочником заканчивается, то необходимо сохранить справочник.

5.3 СОЗДАНИЕ РЕЕСТРА

5.3.1 НАСТРОЙКА СВОЙСТВ РЕЕСТРА

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для создания реестра и настройки его свойств необходимо:

- 1. Выполнить вход в раздел модель данных.
- 2. Нажать кнопку добавить справочник/реестр.
- 3. Пользователь начинает работу во вкладке свойства.

Основные				
Имя:	Логическое	Отображаемое		
Описание сущности:				
Тип сущности:	Реестры			

Продвинутые

Отображение статистики для оператора данных: 🗌

- 4. Выбрать, сущность какого типа необходимо создать: реестр или справочник.
- 5. Задать системное имя реестра, допускается использовать только латиницу и цифры.
- 6. Задать отображаемое имя реестра, это то имя реестра, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Рекомендуется использовать человеко-идентифицируемое название, например, контрагенты или сотрудники.
- 7. Задать описание это реестра, которое используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени реестра недостаточно понятно, какие объекты содержатся в данном реестре, то рекомендуется заполнить описание реестра.
- Отображение статистки для оператора данных. Дополнительная настройка, которая не влияет на метамодель. Если Отображение статистики включено, то на главной странице обработки данных появится блок статистики по данному реестру, где будет указано количество записей, ошибок и потенциальных дубликатов.

• РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

Результатом выполнения данной операции является реестр готовый к моделированию.

5.3.2 НАСТРОЙКА МОДЕЛИ АТРИБУТОВ

НАСТРОЙКА ПРОСТОГО АТРИБУТА

Для настройки простого атрибута выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит к вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.

÷	- 🗁 Держатель Лицензии	Имя	
	— 🕒 Номер лицензии (string, обязательный)	Отображаемое	
	- Ch № (string)	Описание	
	— 🕒 Короткое имя (string, обязательный)		
	— 🕒 Полное имя (string, обязательный)	Обязательный	да
	— 🕒 Руководитель (string)		
	— 🕒 ИНН (string, обязательный)	Поисковое	нет
	— 🕒 OFPH (string, обязательный)	Отображаемое	нет
	— 🕒 ОКПО (string, обязательный)	Гл отображае	NOT
	— 🕒 Исходное имя (string)	11.0100paxae	
	— 🕒 Дата принятия решения о предоставлении лицензии (date, обязательный)	T	
	— 🕒 Дата начала действия лицензии (date, обязательный)	тип атриоута	
	— 🕒 Правовая форма (спр.: "LegacyForm")		
	— 🕒 Сведения о деятельности (спр.: "Activities")		
	- С Адрес (мин: 1, макс: 1)		
	— 🗁 Телефоны (мин: 1, макс: 2, ключ: "type")		
	- Сведения (мин: 1, макс: 1)		
	🕒новый атрибут		

- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.
- 10. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.
- 11. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.

- 12. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному реестру появится поле(-я) для поиска по данному атрибуту.
- 13. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 14. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятный идентификатор, например, название компании, а не её ИНН.
- 15. Задается типа атрибута простой.
- 16. Указывается тип значения простого атрибута. Значение может быть текст, дата, целочисленное, численное значение, логическое значение, бинарные данные, большое текстовое значение.
- 17. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

НАСТРОЙКА АТРИБУТА ТИПА ПЕРЕЧИСЛЕНИЕ

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит ко вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.
- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 18. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 19. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.
- 20. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.

- 8. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному реестру появится поле(-я) для поиска по данному атрибуту.
- 10. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 11. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятные идентификатор, например, название компании, а не её ИНН.
- 12. Задается типа атрибута перечисление. Само перечисление нужно создать до этого момента.
- 13. Выбирается перечисление из списка перечислений.
- 14. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

• НАСТРОЙКА АТРИБУТА ТИПА ССЫЛКА НА ВЕБ-РЕСУРС

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит ко вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.
- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.
- 10. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.

- 11. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- 12. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному реестру появится поле(-я) для поиска по данному атрибуту.
- 13. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 14. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятные идентификатор, например, название компании, а не её ИНН.
- 15. Задается тип атрибута ссылка на веб-ресурс.
- 16. Вводится URL в виде <u>http://www.example.ru/</u>
- 17. Вводится имя атрибута, который будет использоваться в качестве переменной для указания конкретной сущности на веб-ресурсе, в виде {attr_name}.
- 18. Итоговый результат шаблон вида <u>http://www.example.ru/{attr_name}</u>
- 19. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

НАСТРОЙКА АТРИБУТА ТИПА ССЫЛКА НА СПРАВОЧНИК

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит ко вкладке атрибуты.
- 4. Нажимает кнопку создать простой атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- 6. Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, номер ИНН или название организации.
- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "исходное имя" - описание должно содержать следующую информацию "необработанное значение наименования организации, приходящее нам из сторонней системы".
- 8. Задается параметр того, является ли этот атрибут обязательным. Если флаг «обязательный» установлен, то запись пустым обязательным атрибутом будет отклонена.
- 9. Задается параметр того, доступен ли атрибут только на чтение. Если флаг установлен, то пользователь не может редактировать значение атрибута.

- 10. Задается параметр того, скрытый ли атрибут. Если флаг установлен, то атрибут не отображается в общем представлении записи.
- 11. Задается параметр общего поиска по данному атрибуту. Если данный параметр включен, то поле участвует в общем поиске.
- 12. Задается параметр параметрического поиска. Если данный параметр включен, то поле участвует в параметрическом поиске, и в панели параметрического поиска по данному реестру появится поле(-я) для поиска по данному атрибуту.
- 13. Задается параметр отображаемый атрибут. В общем случае, рекомендуется выставлять отображаемыми атрибутами человеко-понятные атрибуты, например, ФИО или должность, не рекомендуется выставлять числовые параметры или идентификаторы.
- 14. Задается параметр главный отображаемый атрибут. В общем случае, рекомендуется выставлять главным отображаемым атрибутом человеко-понятные идентификатор, например, название компании, а не её ИНН.
- 15. Задается типа атрибута ссылка на справочник. Внимание справочник должен быть уже создан.
- 16. Выбирается справочник из списка справочников. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

НАСТРОЙКА КОМПЛЕКСНОГО АТРИБУТА

- 1. Пользователь заходит в раздел модель данных.
- 2. Нажимает кнопку добавить справочник/реестр.
- 3. Пользователь переходит ко вкладке атрибуты.
- 4. Нажимает кнопку создать комплексный атрибут.
- 5. Задается имя атрибута, допускается использовать только латиницу и цифры.
- Задается отображаемое имя атрибута, это то имя атрибута, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов. Крайне рекомендуется использовать человеко-идентифицируемое название, например, адрес.
- 7. Задается описание это атрибута, используется только внутри интерфейсов администратора данных. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из отображаемого имени атрибута недостаточно понятно, какое значение должен иметь данный атрибут, то рекомендуется заполнить описание атрибута, например, название атрибута "адрес" - описание должно содержать следующую информацию "список адресов сотрудника".
- 8. Указывается минимальное и максимальное количество вложенных сущностей.
- 9. Выбирается эта вложенная сущность.
- 10. Выполняется цикл, до создание всех необходимых атрибутов вложенной сущности:
- 11. Нажимается кнопка создать простой атрибут.
- 12. Указываются параметры атрибута, по схеме, описанной выше.

- 13. Указывается ключ, значение атрибута вложенной сущности, ключ может быть указан только после создания атрибутов вложенной сущности. Значение ключа является уникальным, и не может повторяться во вложенных сущностях этой записи. Например, у комплексного атрибута «адрес», ключом может быть атрибут тип адреса – адрес регистрации, домашний, рабочий и т.п. К
- 14. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

5.3.3 НАСТРОЙКА СВЯЗЕЙ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

- 1. Для настройки связей выполняются следующие действия:
- 2. Пользователь заходит в раздел модель данных.
- 3. Выбирает реестр, для которого нужно настроить связь.
- 4. Пользователь переходит к вкладке связи.

	۲	Пехнико-экономические паспор ×					
	Реестр	🔅 Свойства 🔳 Атрибуты 💊 С	вязи 📥 Модель 🔮 Ка	ачество данных 🛛 🙆 Дубликаты	🦏 Консолидация		
	ры/Сп	+ ×	Имя связи:	documents	Обязательная:		
¢	равоч	Нормативные документы Включение	Отображаемое имя:	Нормативные документы	Связной реестр:	Обслуживающие организации	
≔	ники	Обслуживающая Огранизация Ссылка	Тип связи:	Вилюцение			
Ŧ			The object.				
							Û

- 5. Нажимает кнопку добавить связь или выбирает существующую связь, представленную в списке.
- 6. Заполняет свойства связи:
 - а. Имя связи. Допускается использовать только латиницу и цифры. Рекомендуется использовать человеко-понятное наименование связи;
 - Отображаемое имя это то имя связи, которое будет отображаться пользователям системы. Оно может состоять из латинских и кириллических символов, а также пробелов;
 - с. Флаг Обязательная. Указание для механизма валидации.
 - d. Связанный реестр. Выбор реестра, с которым устанавливается связь.
 - е. Тип связи. На данный момент в системе поддерживается два типа связей: "включение" и "ссылка".
- 7. Настраивает атрибуты связи по аналогии с атрибутами реестра/справочника.
- 8. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

5.3.4 НАСТРОЙКА ПРАВИЛ КАЧЕСТВА ДАННЫХ

НАСТРОЙКА ПРАВИЛ ВАЛИДАЦИИ

Для настройки правил валидации выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке качество данных.
- 4. Нажимаем кнопку добавить правило.
- 5. Вводит имя правила. Допускается использовать только латиницу и цифры. Рекомендуется использовать человеко-понятное наименование правила.
- 6. Вводит описание правила. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из имени правила и выбранной функции недостаточно для понимания, того какое действие выполняет данное правило, то рекомендуется заполнить описание правила, например, правило "проверка ИНН" описание должно содержать следующую информацию "проверяется контрольная сумма в значении ИНН", это важно так как ИНН можно проверить и другим способом.
- 7. Выбираем выполняемую функцию. Перед созданием правил качества данных рекомендуется ознакомиться с возможностями функций.
- 8. Выбирается режим, в котором будет работать правило валидация, обогащение или валидация и обогащение. В нашем случае мы выбираем валидацию. Будьте внимательны большинство функции работают только в одном режиме и выбор режима несоответствующего функции делает правило лишенным смысла. Например, функция проверки контрольной суммы ИНН не может работать в режиме обогащения.
- 9. Выбираются источники данных, к которым применяет данное правило качества данных. Так как в один в один реестр нам могут поступать данные из разных источников, то к ним могут применяться разные правила качества данных, например, дата в одном источника хранится как MM/DD/YYYY, а в другом DD.MM.YYYY, для корректной обработки дат из разных систем мы должны привести их к одному формату.
- 10. Следующий этап это маппинг (сопоставление) атрибутов реестра и портов функции.
- 11. Необходимо выбрать на основании какого порта чего создается ошибка. Необходимо выбрать логический порт. В том случае если в порт приходит значение false то, это значит, что валидационное правило выдало ошибку.
- 12. Выбирается фаза обработки данных. Если выбрана фаза до сохранения исходных данных, то при попытке сохранения записи, запись с ошибкой в данном правиле сохранена не будет. Если выбрана фаза после сохранения, то запись с ошибкой будет сохранена, и будет выставлен "флаг" ошибочной записи.
- 13. Выбирается текст сообщения, есть возможность транслировать значение порта функции, или ввести текст самостоятельно. Большинство простых функций не могут самостоятельно выдать текст об ошибке, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать текст об ошибке, это необходимо, так как возможны различные ошибки и тексты сообщения должны отличаться в разных случаях.

- 14. Выбирается критичность ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно выдать критичность ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать критичность ошибки, это необходимо, так как возможны различные ошибки и критичность ошибки должна отличаться при разных ошибках.
- 15. Выбирается категория ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно определить категорию ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут передавать категорию ошибки, это необходимо, так как возможны различные ошибки и категории ошибок будут разные.
- 16. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

НАСТРОЙКА ПРАВИЛ ОБОГАЩЕНИЯ

Для настройки правил валидации и обогащения выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке качество данных.
- 4. Нажимаем кнопку добавить правило.
- 5. Вводит имя правила допускается использовать только латиницу и цифры. Рекомендуется использовать человеко-понятное наименование правила.
- 6. Вводит описание правила. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из имени правила и выбранной функции недостаточно для понимания, того какое действие выполняет данное правило, то рекомендуется заполнить описание правила, например, правило "проверка ИНН" описание должно содержать следующую информацию "проверяется контрольная сумма в значении ИНН", это важно так как ИНН можно проверить и другим способом.
- 7. Выбираем выполняемую функцию. Перед созданием правил качества данных рекомендуется ознакомиться с возможностями функций.
- 8. Выбирается режим, в котором будет работать правило валидация, обогащение или валидация и обогащение. В нашем случае мы выбираем обогащение. Будьте внимательны большинство функции работают только в одном режиме и выбор режима несоответствующего функции делает правило лишенным смысла. Например, функция проверки контрольной суммы ИНН не может работать в режиме обогащения.
- 9. Выбираются источники данных, к которым применяет данное правило качества данных. Так как в один в один реестр нам могут поступать данные из разных источников, то к ним могут применяться разные правила качества данных, например, дата в одном источника хранится как MM/DD/YYYY, а в другом DD.MM.YYYY, для корректной обработки дат из разных систем мы должны привести их к одному формату.

- 10. Следующий этап это маппинг (сопоставление) атрибутов реестра и портов функции.
- 11. Выбирается фаза обработки данных.
- 12. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

НАСТРОЙКА ПРАВИЛ ВАЛИДАЦИИ И ОБОГАЩЕНИЯ

Для настройки правил валидации и обогащения выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке качество данных.
- 4. Нажимаем кнопку добавить правило.
- 5. Вводит имя правила допускается использовать только латиницу и цифры. Рекомендуется использовать человеко-понятное наименование правила.
- 6. Вводит описание правила. Оно может состоять из латинских и кириллических символов, а также пробелов. В том случае, если из имени правила и выбранной функции недостаточно для понимания, того какое действие выполняет данное правило, то рекомендуется заполнить описание правила, например, правило "проверка ИНН" описание должно содержать следующую информацию "проверяется контрольная сумма в значении ИНН", это важно так как ИНН можно проверить и другим способом.
- 7. Выбираем выполняемую функцию. Перед созданием правил качества данных рекомендуется ознакомиться с возможностями функций.
- 8. Выбирается режим, в котором будет работать правило валидация, обогащение или валидация и обогащение. В нашем случае мы выбираем валидацию и обогащение. Данный выбор означает, что функция может работать в двух режимах, и в зависимости от данных. Если в данных содержится ошибка, то отработает только валидационная часть функции, если ошибки нет, отработает валидационная и обогащающая часть функции.
- 9. Выбираются источники данных, к которым применяет данное правило качества данных. Так как в один в один реестр нам могут поступать данные из разных источников, то к ним могут применяться разные правила качества данных, например, дата в одном источника хранится как MM/DD/YYYY, а в другом DD.MM.YYYY, для корректной обработки дат из разных систем мы должны привести их к одному формату.
- 10. Следующий этап это маппинг (сопоставление) атрибутов реестра и портов функции.
- 11. Необходимо выбрать на основании какого порта чего создается ошибка. Необходимо выбрать логический порт. В том случае если в порт приходит значение false то, это значит, что валидационное правило выдало ошибку.
- 12. Выбирается фаза выполнения валидационной составляющей функции. Если выбрана фаза до сохранения исходных данных, то при попытке сохранения записи, запись с ошибкой в данном правиле сохранена не будет. Если выбрана фаза после сохранения, то запись с ошибкой будет сохранена, и будет выставлен "флаг" ошибочной записи.
- 13. Выбирается текст сообщения, есть возможность транслировать значение порта функции, или ввести текст самостоятельно. Большинство простых функций не могут самостоятельно

выдать текст об ошибке, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать текст об ошибке, это необходимо, так как возможны различные ошибки и тексты сообщения должны отличаться в разных случаях.

- 14. Выбирается критичность ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно выдать критичность ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут транслировать критичность ошибки, это необходимо, так как возможны различные ошибки и критичность ошибки должна отличаться при разных ошибках.
- 15. Выбирается категория ошибки, есть возможность транслировать значение порта функции, или ввести значение самостоятельно. Большинство простых функций не могут самостоятельно определить категорию ошибки, так как они не владеют контекстом выполнения, с другой стороны композитные и сторонние функции могут передавать категорию ошибки, это необходимо, так как возможны различные ошибки и категории ошибок будут разные.
- 16. Выбирается фаза выполнения обогащающей составляющей функции.
- 17. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

5.3.5 НАСТРОЙКА ПРАВИЛ ПРОВЕРКИ НА ДУБЛИКАТЫ

• ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для настройки правил проверки на дубликаты выполняются следующие действия:

- 1. Пользователь заходит в раздел модель данных.
- 2. Открывает справочник/реестр.
- 3. Пользователь переходит ко вкладке дубликаты.
- 4. Выбирает ручное или автоматическое правило поиска дубликатов
- 5. Следующий этап это маппинг (сопоставление) атрибутов реестра и портов функции. Выбор тех атрибутов, которые будут проверяться на совпадение.
- 6. Если на этом работа с реестром заканчивается, то необходимо сохранить реестр.

6 ФУНКЦИИ

6.1 НАСТРОЙКА КОМПОЗИТНЫХ ФУНКЦИЙ

В системе UniData композитные функции создаются посредством графического редактора. В графическом редакторе пользователь выполняет следующие операции:

СОЗДАНИЕ ВХОДНЫХ ПОРТОВ – порт из которого функции будет использовать значения для работы. Настраивается имя порта, его тип, описание, и признак обязательности.

СОЗДАНИЕ ВЫХОДНЫХ ПОРТОВ – порт из которого функции будет использовать значения для работы. Настраивается имя порта, его тип, описание, и признак обязательности.

ДОБАВЛЕНИЕ ФУНКЦИЙ - добавление функций, которые будут находиться в составе композитной функции.

МАППИНГ ФУНКЦИЙ — связь портов композитной функции с портами вложенных функций, и, при необходимости, связывание портов функций между собой.

6.1.1 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

- 1. Необходимо зайти в раздел функции.
- 2. Нажать кнопку добавить функцию.
- 3. Выбрать вариант композитная функция.
- 4. Настроить входные порты. Необходимо указать имя порта и его тип, дополнительно можно указать описание и обязательность порта. Внимание, если в порт, отмеченный обязательным не попадут данные, то функция выдаст ошибку.
- 5. Настроить выходные порты.
- 6. Добавить вложенные функции.
- 7. Выполнить маппинг.
- 8. Проверить работу композитной функции.
- 9. Сохранить.

6.2 СОЗДАНИЕ СТОРОННИХ ФУНКЦИЙ

6.2.1 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

- 1. Необходимо зайти в раздел функции.
- 2. Нажать кнопку добавить функцию.
- 3. Выбрать вариант сторонняя функция.
- 4. Нажать кнопку загрузить функцию.
- 5. Выбрать JAR файл. В JAR файле должен быть только один метод, в противном случае система, выдаст ошибку.
- 6. Дождитесь окончания загрузки файла.

6.2.2 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

Функция, готовая для использования в правилах качества данных и в композитных функциях.

6.2.3 УСТРАНЕНИЕ ВОЗМОЖНЫХ НАРУШЕНИЙ

ОШИБКА	УСТРАНЕНИЕ
Ошибка загрузки файла	Загрузить JAR файл, соответствующий требованиям.
	Проверить работоспособность браузера.
	Проверить работоспособность соединения с сервером UniData.
	Проверить работоспособность сервера UniData.
Чрезмерно длительная загрузка	Обновить страницу. Повторно авторизоваться. Повторно загрузить функцию в систему.
Отсутствует реакция на нажатие кнопки загрузить	Проверить работоспособность браузера.
	Проверить работоспособность соединения с сервером UniData.
	Проверить работоспособность сервера UniData.
Ошибка доступа	Повторно авторизоваться. Повторно загрузить функцию в систему.
Отображение подтверждения об обновлении функции	В системе уже существует функция с таким название JAVA метода, как в загружаемом файле. Необходимо загрузить JAR файл, соответствующий требованиям, в котором используется уникальное наименование JAVA класса

6.3 РЕДАКТИРОВАНИЕ СТОРОННИХ ФУНКЦИЙ

Редактирование сторонних функций осуществляется двумя способами. Последовательность действий для каждого из них приведена ниже.

6.3.1 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ 1

- 1. Необходимо зайти в раздел функции.
- 2. Нажать кнопку добавить функцию.
- 3. Выбрать вариант сторонняя функция.
- 4. Нажать кнопку загрузить функцию.
- 5. Выбрать JAR файл. JAR файл должен содержать JAVA метод обновляемой функции. В JAR файле должен быть только один метод, в противном случае система, выдаст ошибку.
- 6. Дождитесь окончания загрузки файла.
- 7. Подтвердить обновление функции.
- 8. После чего новая функция будет доступна для использования.

6.3.2 ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ 2

- 1. Необходимо зайти в раздел функции.
- 2. Выбрать папку разное.
- 3. Выбрать необходимую функцию.
- 4. Нажать кнопку загрузить функцию.
- 5. Выбрать JAR файл. JAR файл должен содержать JAVA метод обновляемой функции. В JAR файле должен быть только один метод, в противном случае система, выдаст ошибку.
- 6. Дождитесь окончания загрузки файла.
- 7. Подтвердить обновление функции.
- 8. После чего новая функция будет доступна для использования.

6.3.3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

Функция, готовая для использования в правилах качества данных и в композитных функциях.

6.3.4 УСТРАНЕНИЕ ВОЗМОЖНЫХ НАРУШЕНИЙ

ОШИБКА	УСТРАНЕНИЕ
Ошибка загрузки файла	Загрузить JAR файл, соответствующий требованиям.
	Проверить работоспособность браузера.
	Проверить работоспособность соединения с сервером UniData.
	Проверить работоспособность сервера UniData.
	В случае последовательности выполнения 2, следует проверить соответствие названия метода в файле названию метода в функции.
Чрезмерно длительная загрузка	Обновить страницу. Повторно авторизоваться. Повторно загрузить функцию в систему.
Отсутствует реакция на нажатие кнопки загрузить	Проверить работоспособность браузера.
	Проверить работоспособность соединения с сервером UniData.
	Проверить работоспособность сервера UniData.
Ошибка доступа	Повторно авторизоваться. Повторно загрузить функцию в систему.
Функция не обновлена	Проверить версию загружаемого файла. Проверить соответствие названия метода в файле названию метода в функции. Загрузить JAR файл, соответствующий требованиям.

7 РАБОТА С ДАННЫМИ

7.1 ВВЕДЕНИЕ

В системе UniData все данные хранятся в реестрах или справочниках.

РЕЕСТРЫ — записи с набором атрибутов и их значений, а также модель связей между записями. Так же в модели реестра описываются привила валидации и обогащения данных.

СПРАВОЧНИКИ – набор атрибутов и их значений, основное отличие заключается в том, что к справочникам не применяются правила качества данных и нет связей.

Реестры и справочники состоят из атрибутов и их значений, а также вложенных сущностей.

АТРИБУТЫ – пара ключ-значение, ниже указаны типы и возможные значения атрибутов.

ПРОСТОЙ АТРИБУТ – имя этого атрибута является ключом, ниже перечислены типы возможных значений.

СТРОКОВОЕ – строка без дополнительных ограничений, длина 2048 символов.

ЧЕЛОЧИСЛЕННОЕ – целочисленное значение, длина 2⁶⁴

ЧИСЛЕННОЕ – числовое значение, длина 2⁶⁴

ЛОГИЧЕСКОЕ – может принимать следующие значение: true, false, null.

ДАТА/ВРЕМЯ – дата в формате ISO timestamp

ДАТА – дата

ВРЕМЯ - время

БИНАРНЫЙ МАССИВ — значением это приложенный файл, поиск по которому не выполняется.

ТЕКСТОВЫЙ МАССИВ – массив текстовых данных, хранящийся в базе данных

ССЫЛКА НА СПРАВОЧНИК – указывается конкретный справочник, в котором будет искаться значение атрибута.

ПЕРЕЧИСЛЕНИЕ – простой список возможных значений атрибута.

ССЫЛКА НА ВЕБ-РЕСУРС — шаблон ссылки на сторонний ресурс (адрес и переменная), в записи из шаблона формируется ссылка с использованием данных записи.

ВЛОЖЕННЫЕ СУЩНОСТИ — набор сущностей, вложенных в одну запись. Например, сущность номер телефона с атрибутами: тип, код города, номер, является вложенной к записи сотрудник.

Сотрудник может иметь несколько телефонов: {домашний, +7495,6543219} и {мобильный, +7999,6543219}.

Записи в реестре могут подвергаться определенным проверкам и трансформациям.

ПРАВИЛА КАЧЕСТВА ДАННЫХ — набор функций, связанный с атрибутами, и предназначенный либо для валидации данных, либо для их обогащения.

ПРОВЕРКА НА ДУБЛИКАТЫ — настройка правил проверки на наличие дубликатов, подразумевается полное соответствие записей, по заданным в правилах, атрибутах.

ССЫЛКИ – связь одной записи с другой записью в реестре.

7.2 РАЗДЕЛ ОБРАБОТКА ДАННЫХ

Основное предназначение раздела — это работа с данными. Этот раздел состоит из трех экранов: Главная, Данные и Импорт / Экспорт.

ГЛАВНАЯ – стартовая страница для специалиста по обработке данных, здесь отображается сводка по реестрам. По каждому реестру отображается количество записей, количество новых записей, количество обновленных записей, количество записей с ошибками, количество потенциальных дубликатов. Кроме этого отображаются графики изменения этих показателей по времени. И есть диаграмма критичностей имеющихся ошибок. Все эти инструменты предназначены для аналитики и планирования предстоящей работы.

ДАННЫЕ — экран для работы с поиском и записями. Основной рабочий экран специалиста по обработке данных, в котором осуществляются все операции по работе с записями. Экран состоит и з четырёх областей: области навигации, области поиска, области результатов поиска и области записей.

ИМПОРТ / ЭКСПОРТ – экран для выгрузки данных их системы UniData и для загрузки данных в систему UniData.

7.2.1 ОБЩАЯ СХЕМА РАБОТЫ С ДАННЫМИ

- Для начала работы необходимо определить выполняемую задачу, если из сторонних источников заранее известна запись или группа записей, то нужно перейти к следующему пункту. В том случае, когда точный объект не известен, то нужно перейти в экран Главная и выявить реестры в которых есть данные с ошибкам или потенциальные дубликаты.
- 2. Необходимо найти запись с помощью панели поиска и выбрать её.
- 3. Проделать необходимые изменение в окне с записью.
- 4. Сохранить изменения.

7.3 ПОИСК ЗАПИСИ



Для поиска необходимо сначала выбрать реестр или справочник, в котором будет осуществляться поиск по записям.

После чего можно нажать кнопку все и отобразятся все записи реестра. Альтернативными условиями являются введение текста в текстовое поле, поиск по записям с ошибками качества данных, поиск по записям являющимся потенциальными дубликатами и параметрический поиск.

ПОИСКОВОЕ УСЛОВИЕ	ПРИЦИП ФОРМИРОВАНИЯ ПОИСКОВОЙ ВЫДАЧИ
BCE	Все записи, находящиеся в данном справочнике или реестре
ЗНАЧЕНИЕ В ТЕКСТОВОМ ПОЛЕ	Записи, содержащие значение из текстового поля минимум в одном из атрибутов типа текст, дата, целочисленное значение, численное значение, указанным как поисковые.
ТОЛЬКО С ОШИБКАМИ	Записи реестра содержащие ошибки качества данных
ТОЛЬКО ДУБЛИКАТЫ	Записи реестра имеющие потенциальные дубликаты

ПОИСКОВОЕ УСЛОВИЕ	ПРИЦИП ФОРМИРОВАНИЯ ПОИСКОВОЙ ВЫДАЧИ				
ПАРАМЕТРИЧЕСКИЙ	Записи, содержащие текстовое значение из текстового поля в				
ПОИСК (ТЕКСТОВЫЙ)	указанном атрибуте.				
ПАРАМЕТРИЧЕСКИЙ ПОИСК (ЧИСЛЕННЫЙ ТОЧНЫЙ)	Записи, содержащие численное значение из текстового поля в указанном атрибуте.				
ПАРАМЕТРИЧЕСКИЙ	Записи, значение атрибута которых находится в заданном				
ПОИСК (ЧИСЛЕННЫЙ	диапазоне. Крайние значения диапазона включаются в				
ИНТЕРВАЛЬНЫЙ)	поисковую выдачу.				
ПАРАМЕТРИЧЕСКИЙ	Записи, логическое значение которых равно указанному				
ПОИСК (ЛОГИЧЕСКИЙ)	значению атрибута.				

7.3.1 ДОПОЛНИТЕЛЬНЫЕ УСЛОВИЯ ПРИ ФОРМИРОВАНИИ ПОИСКОВОЙ ВЫДАЧИ

- 1. Запись должна отвечать одновременно всем указанным поисковым условиям.
- 2. Запись должна иметь период актуальности, указанный интервал которого включает дату и время поискового запроса.
- 3. Запись не должна быть помечена как удаленная.

7.4 CRUD ЗАПИСИ

7.4.1 СОЗДАНИЕ ЗАПИСИ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для создания записи необходимо:

- 1. Перейти в экран данные.
- 2. Выбрать реестр или справочник, в котором необходимо создать новую запись.
- 3. Нажать кнопку добавить новую запись.
- 4. Ввести значения атрибутов. Для сохранения записи необходимо, что бы все введенные значение соответствовали типу атрибута и обязательные атрибуты имели значения.
- 5. Сохранить запись.

• РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

В том случае если все необходимые данные заполнены корректно, отобразится уведомление о сохранении данных. В остальных случаях перейдите к диагностике возможных нарушений.

ДИАГНОСТИКА И УСТРАНЕНИЕ ВОЗМОЖНЫХ НАРУШЕНИЙ

ОШИБКА	УСТРАНЕНИЕ		
Форма не валидна	Заполнить все обязательные поля.		
	Корректировать значения атрибутов, в соответствии с типом. Атрибуты с ошибочными значениями находятся в состоянии ошибки.		
	Корректировать значения атрибутов, проверяемых правилами качествам данных. Атрибуты с ошибочными значениями находятся в состоянии ошибки.		
Отсутствует реакция на нажатие кнопки сохранить	Проверить работоспособность браузера.		
	Проверить работоспособность соединения с сервером UniData		
	Проверить работоспособность сервера UniData		
Ошибка доступа	Повторно авторизоваться. Внести данные в систему.		

7.4.2 ПРОСМОТР ЗАПИСИ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для просмотра записи необходимо:

- 1. Перейти в экран данные.
- 2. Выбрать реестр или справочник, в котором находится необходимая для просмотра запись.
- 3. Выполнить поиск минимум по одному из условий.
- 4. Выбрать необходимую запись в списке результатов поиска.
- 5. Просмотреть запись.

ДИАГНОСТИКА И УСТРАНЕНИЕ ВОЗМОЖНЫХ НАРУШЕНИЙ

ОШИБКА	УСТРАНЕНИЕ
Запись отсутствует в списке результатов поиска	Проверить корректность введенных поисковых условий.
	Выполнить поиск по неактуальным записям
	Выполнить поиск по удаленным записям
	Проверить наличие доступа к запрашиваемым данным

ОШИБКА	УСТРАНЕНИЕ			
Отсутствует реакция на нажатие кнопки поиск	Проверить работоспособность браузера.			
	Проверить работоспособность соединения с сервером UniData			
	Проверить работоспособность сервера UniData			
Ошибка доступа Повторно авторизоваться. Повторить поисков				

7.4.3 РЕДАКТИРОВАНИЕ ЗАПИСИ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для редактирования записи необходимо:

- 1. Перейти в экран данные.
- 2. Выбрать реестр или справочник, в котором находится необходимая для редактирования запись.
- 3. Выполнить поиск минимум по одному из условий.
- 4. Выбрать необходимую запись в списке результатов поиска.
- 5. Внести корректировки. Для сохранения записи необходимо, что бы все введенные значение соответствовали типу атрибута и обязательные атрибуты имели значения.
- 6. Сохранить.

• РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

В том случае если все необходимые данные заполнены корректно, отобразится уведомление о сохранении данных. В остальных случаях перейдите к диагностике возможных нарушений.

ОШИБКА	УСТРАНЕНИЕ
Запись отсутствует в списке	Проверить корректность введенных поисковых условий.
результатов поиска	Выполнить поиск по неактуальным записям
	Выполнить поиск по удаленным записям
	Проверить наличие доступа к запрашиваемым данным
Отсутствует реакция на	Проверить работоспособность браузера.
нажатие кнопки поиск	Проверить работоспособность соединения с сервером UniData
	Проверить работоспособность сервера UniData
Форма не валидна	Заполнить все обязательные поля.
	Корректировать значения атрибутов, в соответствии с типом. Атрибуты с ошибочными значениями находятся в состоянии ошибки.

♦ УСТРАНЕНИЕ ВОЗМОЖНЫХ НАРУШЕНИЙ

ОШИБКА	УСТРАНЕНИЕ			
	Корректировать значения атрибутов, проверяемых правилами качествам данных. Атрибуты с ошибочными значениями находятся в состоянии ошибки.			
Отсутствует реакция на нажатие кнопки сохранить	Проверить работоспособность браузера.			
	Проверить работоспособность соединения с сервером UniData			
	Проверить работоспособность сервера UniData			
Ошибка доступа	Повторно авторизоваться. Внести данные в систему.			

7.4.4 УДАЛЕНИЕ ЗАПИСИ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для удаления записи необходимо:

- 1. Перейти в экран данные
- 2. Выбрать реестр или справочник, в котором находится необходимая для удаления запись.
- 3. Выполнить поиск минимум по одному из условий.
- 4. Выбрать необходимую запись в списке результатов поиска.
- 5. Проверить отсутствие связи записи с другими записями.
- 6. Удалить

• РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

Запись будет помечена как удаленная и не будет отображаться в поиске по записям.

УСТРАНЕНИЕ ВОЗМОЖНЫХ НАРУШЕНИЙ

ОШИБКА	УСТРАНЕНИЕ
Запись отсутствует в списке результатов поиска	Проверить корректность введенных поисковых условий.
	Выполнить поиск по неактуальным записям
	Выполнить поиск по удаленным записям
	Проверить наличие доступа к запрашиваемым данным
Отсутствует реакция на нажатие кнопки удалить	Проверить работоспособность браузера.
	Проверить работоспособность соединения с сервером UniData
	Проверить работоспособность сервера UniData
Ошибка доступа	Повторно авторизоваться. Повторить поисковый запрос

7.5 РАБОТА С ЗАПИСЯМИ

7.5.1 РАБОТА С ОШИБКАМИ КАЧЕСТВА ДАННЫХ

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для работы с ошибками качества данных необходимо:

- 1. Перейти в экран данные.
- 2. Выбрать реестр или справочник, в котором находится необходимая для редактирования запись.
- 3. Выполнить поиск минимум по одному из условий. Для скрытия записей без ошибок, необходимо поставить поисковое условие только с ошибками.
- 4. Выбрать необходимую запись в списке результатов поиска.
- 5. Внести корректировки. Для сохранения записи необходимо, что бы все введенные значение соответствовали типу атрибута и обязательные атрибуты имели значения.
- 6. Сохранить.

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

В том случае, если изменения были корректными, то ошибка будет устранена. Если данная запись более не содержит ошибок, то она перестанет помечаться ошибочной.

ДИАГНОСТИКА И УСТРАНЕНИЕ ВОЗМОЖНЫХ НАРУШЕНИЙ

ОШИБКА	УСТРАНЕНИЕ
Запись отсутствует в списке результатов поиска	Проверить корректность введенных поисковых условий.
	Выполнить поиск по неактуальным записям
	Выполнить поиск по удаленным записям
	Проверить наличие доступа к запрашиваемым данным
Отсутствует реакция на нажатие кнопки поиск(все)	Проверить работоспособность браузера.
	Проверить работоспособность соединения с сервером UniData
	Проверить работоспособность сервера UniData
Форма не валидна	Заполнить все обязательные поля.
	Корректировать значения атрибутов, в соответствии с типом. Атрибуты с ошибочными значениями находятся в состоянии ошибки.

ОШИБКА	УСТРАНЕНИЕ			
	Корректировать значения атрибутов, проверяемых правилами качествам данных. Атрибуты с ошибочными значениями находятся в состоянии ошибки.			
Отсутствует реакция на нажатие кнопки сохранить	Проверить работоспособность браузера.			
	Проверить работоспособность соединения с сервером UniData			
	Проверить работоспособность сервера UniData			
Ошибка доступа	Повторно авторизоваться. Внести данные в систему.			

7.5.2 ОБЪЕДИНИИЕ ЗАПИСЕЙ

НАЗНАЧЕНИЕ, ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ

В процессе работы, пользователь может принимать решения о консолидации нескольких эталонных записей одного реестра или справочника в одну с целью избавления от дубликатов. Система позволяет консолидировать произвольное количество эталонных записей за один раз, при этом пользователь всегда выбирает запись победитель, а все остальные выбранные записи автоматически назначаются проигравшими. В результате консолидации в системе остаётся одна эталонная запись (победителя), включающая в себя все исходные записи как победителя, так и проигравших. Проигравшие эталонные записи исключаются из дальнейшего взаимодействия (помечаются специальным образом во внутреннем хранилище Системы).

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

Для объединения записей выполняются следующие действия:

- 1. Пользователь выбирает эталонную запись, которая должна стать победителем, использую стандартные поисковые возможности.
- 2. В окне просмотра деталей записи, пользователь нажимает кнопку "Объединить".
- В результате чего, активная запись переводится в специальный режим выбора дубликатов. При этом основная закладка переименовывается в "Дубликаты (N)", где отображает количество набранных записей для консолидации.
- 4. Находясь в режиме выбора дубликатов, пользователь должен добавить все другие записи, требующие консолидации:
 - а. Для этого каждую такую запись нужно сначала открыть на просмотр, используя стандартный функционал поиска.
 - b. После этого нажать кнопку "Объединить".
 - с. В результате выбранная запись будет добавлена внутрь основной закладки "Дубликаты (N)", при этом запись исчезнет из списка открытых записей.

- 5. После того, как все записи требующие объединения выбраны, пользователь возвращается на закладку "Дубликаты (N)" для финального подтверждения действия:
 - а. Для этого пользователь должен подтвердить выбор каждого кандидата на объедение путём нажатия путём нажатия на checkbox, над колонкой с деталями записи.
 - b. После этого пользователь нажимает кнопку "Объединить записи".
- 6. В результате все выбранный записи консолидируются в один эталон, содержащий все исходные записи.

7.5.3 РАБОТА С ЗАПИСЯМИ, ЯВЛЯЮЩИМИСЯ ПОТЕНЦИАЛЬНЫМИ ДУБЛИКАТАМИ

После загрузки данных, часть записей справочников или реестров может быть идентифицировано системой, как потенциальные дубликаты. В этом случае оператор данных должен принимать решение о фактической консолидации эталонных записей. Все такие записи естественным образом разбиваются на кластере, где внутри каждого кластера все записи являются потенциальными дубликатами друг друга. При открытие любой записи реестра или справочника, имеющей потенциальные дубликаты, количество таких дублей будет отображаться красным цветом на соответствующей кнопке "Количество дубликатов".

	Unida	ta ×					
۲	> C	localhost:1841			🌆 🔶 🚥	4 🔏 🛡	⊙ ≡
	۲	Результаты (521) 🔇 <table-cell-rows> 🖺</table-cell-rows>	СПб ГБУЗ"Городская поли	клиника $^{\times}$ СПб ГБУЗ"Городская поликлиника $^{\times}$	УФСБ России по	г. Санкт-Петерб	y × >
	Тоиск	≪ < стр. <mark>52</mark> of 53 > »>	Держатель Лицензии Связи	Исходные записи Ошибки			
		СПБ ГБУЗ ГКОД		🗌 Показывать скрытые атрибуты			
¢		САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕН	Номер лицензии:	ЛО-78-04-000073			*
		ГБУЗ"Городская поликлиника №17"	N≘∶				Ŵ
¥		7806044912	Короткое имя:	УФСБ России по г. Санкт-Петербургу и Ленинградской обла	сти		4
		САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕН	Полное имя:	УПРАВЛЕНИЕ ФЕДЕРАЛЬНОЙ СЛУЖБЫ БЕЗОПАСНОСТИ	РОССИЙ(6
.		ГБУЗ"Городская поликлиника №17"	Руководитель:			Количество ду	бликатов
			ИНН:	7825007923			
, the second sec			OFPH:	1027809230470			
Ē		УФСБ России по г. Санкт-Петербургу и Ле	ОКПО:	7569173			
¢°		7825007923 УПРАВЛЕНИЕ ФЕДЕРАЛЬНОЙ СЛУЖБЫ БІ	Исходное имя:	Управление Федеральной службы безопасности Российской	й Федераці		
			Дата принятия решения о предоставлении	2012-12-05			
		СПб ГБУЗ"Городская больница №15" 7907015216	лицензии:				
		САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕН	дата начала деиствия лицензии:	2012-12-05			
4			Правовая форма:		~		
		СПб ГБУЗ "ГТБ №2" 7802041389	Сведения о деятельности:		~		
		САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕН	х Алрес				
			+ vithee	Фобавить запись			
		спотъуз породская оольница №38 им. 7820013553					
		САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕН	✓ Сведения				

По нажатию кнопки пользователь переходит в режим сравнения потенциальных дубликатов и принятия финального решения о выборочной консолидации в единую эталонную запись. Все дальнейшие действия полностью повторяют раздел "6.5.2 Объединение записей", с то лишь разницей, что пользователю не нужно самому искать дубликаты, так как Системы сделала этот шаг за него.

7.5.4 ПРОСМОТР ИСХОДНЫХ ВЕРСИЙ ЗАПИСИ

Любая эталонная запись состоит справочника или реестра состоит из одной или более исходных записей. Находясь в режиме просмотра деталей конкретной эталонной записи, пользователь может выбрать закладку второго уровня – "Исходные записи" и переключится в режим отображения исходных записей, каждая из которых представлена отдельной колонкой, содержащей все необходимые детали:

	۲	۲	500 Среднегаванский пр. \times					
~	Поиск	Резуль	Технико-экономические паспо	Связи Исходные зап	иси Ошибки			
¢		таты (2)	Источники данных:	unidata			unidata	
:=			🗸 Системные атрибуты					
÷			Улица:	Среднегаванский пр.			test1	4
			Дом:	14			12	
÷			Корпус:				1	
n •			Литера:	A			A	
*			Район:	Василеостровский			Кировский	
Ņ			Серия, тип проекта:	Индивидуальный			Индивидуальный	
œ			Год постройки:	1901			2012	
			Год проведения реконструкции:				2015	
*			Общая площадь здания:	0			23	
- a			Площадь жилых помещений:	142485997559				
			Площадь нежилых помещений функционального назначения:					
			Чиспо этажей:	6				

7.6 ВЕРСИОНИРОВАНИЕ ЗАПИСЕЙ

Создание версии записи может производиться двумя способами с помощью интерфейса UniData или посредством API.

7.6.1 СОЗДАНИЕ НОВОЙ ВЕРСИИ ЗАПИСИ С ПОМОЩЬЮ ИНТЕРФЕЙСА UNIDATA (БЕЗ ПЕРИОДОВ АКТУАЛЬНОСТИ)

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

1. Перейти в экран данные.

- 2. Выбрать реестр или справочник, в котором находится необходимая для редактирования запись.
- 3. Выбрать необходимую запись. При необходимости воспользоваться поиском.
- 4. Изменить необходимые атрибуты записи.
- 5. Сохранить запись

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

Создается новая версия записи от имени системы UniData. В исходных записях добавляется запись. В связи с тем, что система UniData имеет больший вес чем любая другая система, и версия созданных изменений является последней, то эталон записи изменится на только что созданную версию.

7.6.2 СОЗДАНИЕ НОВОЙ ВЕРСИИ ЗАПИСИ С ПОМОЩЬЮ ИНТЕРФЕЙСА UNIDATA (С СОЗДАНИЕМ ПЕРИОДОВ АКТУАЛЬНОСТИ)

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

- 1. Перейти в экран данные.
- 2. Выбрать реестр или справочник, в котором находится необходимая для редактирования запись.
- 3. Выбрать необходимую запись. При необходимости воспользоваться поиском.
- 4. Нажать кнопку создать период актуальности.
- 5. Указать дату окончания периода действия текущей версии записи.
- Указать дату начала действия следующего периода актуальности. Обратите внимание, что периоды не пересекаются, но могут обладать разрывами.
- 7. Указать дату окончания следующего периода актуальности.
- 8. Изменить необходимые атрибуты записи.
- 9. Сохранить.

• РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

У записи появляются периоды актуальности, от двух до трёх периодов, в зависимости от того, какая дата окончания у создаваемого периода. Созданная запись действует на указанном периоде, предыдущая версия действует на остальных периодах актуальности.

7.6.3 СОЗДАНИЕ НОВОЙ ВЕРСИИ ЗАПИСИ С ПОМОЩЬЮ ИНТЕРФЕЙСА UNIDATA (С РЕДАКТИРОВАНИЕМ ПЕРИОДОВ АКТУАЛЬНОСТИ)

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ

- 1. Перейти в экран данные.
- 2. Выбрать реестр или справочник, в котором находится необходимая для редактирования запись.

- 3. Выбрать необходимую запись. При необходимости воспользоваться поиском.
- 4. Нажать кнопку создать период актуальности.
- 5. Указать дату окончания периода действия текущей версии записи.
- 6. Указать дату начала действия следующего периода актуальности. Обратите внимание, что периоды не пересекаются, но могут обладать разрывами.
- 7. Указать дату окончания следующего периода актуальности.
- 8. Изменить необходимые атрибуты записи.
- 9. Сохранить.

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

У отредактированной записи может измениться количество периодов актуальности. Созданная запись действует на указанном периоде, предыдущие версии верны на остальных периодах актуальности.

ПРИЛОЖЕНИЯ

приложение а. перечень терминов и сокращений

ОБОЗНАЧЕНИЕ (СОКРАЩЕНИЕ)	ОПРЕДЕЛЕНИЕ (РАСШИФРОВКА)		
Атрибуты	пара ключ-значение, ниже указаны типы и возможные значения атрибутов		
Валидация	проверка данных. Например, можно проверять равенство атрибутов		
Вес источника	приоритет источника, устанавливаемый записям, пришедшим из этой системы. Варьируется от 0 до 99		
Вложенные сущности	набор сущностей, вложенных в одну запись		
Источник	система, поставщик исходных данных		
Композитные функции	композитные функции представляют из себя последовательность исполняемых функций. Могут состоять из всех типов функций		
Маппинг функций	связь портов композитной функции с портами вложенных функций, и, при необходимости, связывание портов функций между собой		
Обогащение	трансформация или дополнение данных		
Перечисление	простой список возможных значений атрибута		
Перечисления	простой список возможных значений		
Пользователи	пользователь получает права в системе UniData, согласно имеющимся у него ролям. Пользователь обладает уникальным именем в системе - login. Пользователь может обладать более чем одной ролью в системе		
Права	право на просмотр объекта, право на создание, на редактирование и право на удаление объекта. Так же есть полные права, являющиеся совокупностью четырёх предыдущих		
Правила качества данных	набор функций, связанный с атрибутами, и предназначенный либо для валидации данных, либо для их обогащения		
Проверка на дубликаты	настройка правил проверки на наличие дубликатов, подразумевается полное соответствие по записям, по заданным в правилах атрибутам		
Простой атрибут	имя такого атрибута является ключом		
Простые функции	стандартные функции системы UniData. Простые функции могут обработать данные или проверить их, например, можно перевести текстовое поле в верхний регистр		
Реестры	записи с набором атрибутов и их значений, а также модель связей между записями. В модели реестра описываются привила валидации и обогащения данных		
Роли	роль определяет объекты доступа, такие как администрирование системы, администрирование данных, а также права к реестрам		
Список элементов	перечень записей, принадлежащих одному объекту		

ОБОЗНАЧЕНИЕ (СОКРАЩЕНИЕ)	ОПРЕДЕЛЕНИЕ (РАСШИФРОВКА)			
Справочники	набор атрибутов и их значений, основное отличие заключается в том, что к справочникам не применяются правила качества данных и нет связей			
Ссылка на веб- ресурс	шаблон ссылки на сторонний ресурс (адрес и переменная), в записи из шаблона формируется ссылка с использованием данных записи			
Ссылка на справочник	указывается конкретный справочник, в котором будет искаться значение атрибута			
Ссылки	связь одной записи с другой записью в реестре			
Сторонние функции	представляют из себя исполняемый JAR файл, загруженный в систему			
Функция	программный компонент, валидирующий или обогащающий данные			
Бинарный массив	массив двоичных данных, значение которого - приложенный файл, поиск по которому не выполняется			
Логическое	может принимать следующие значение: true, false, null			
Строковый массив	массив текстовых данных, хранящийся в базе данных			
Время	время			
Дата	дата			
Дата/время	дата в формате ISO timestamp			
Целочисленное	целочисленное значение, длина 264			
Численное	числовое значение, длина 264			
Строковое	строка без дополнительных ограничений, длина 2048 символов			

ПРИЛОЖЕНИЕ Б. ПРОСТЫЕ ФУНКЦИИ

НАЗВАНИЕ	ТИП ФУНКЦИИ	ОПИСАНИЕ			
СТРОКОВЫЕ					
УБРАТЬ ПРОБЕЛЫ	обогащение	Убирает пробелы в начале и в конце строки			
УБРАТЬ ЛИШНИЕ ПРОБЕЛЫ	обогащение	Убирает повторяющиеся пробелы			
РЕГУЛЯРНОЕ ВЫРАЖЕНИЕ	обогащение	Выбирает подстроку согласно регулярному выражению			
ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ	обогащение	Устанавливает значение по умолчанию если входной параметр не заполнен			
ФОРМАТИРОВАТЬ	обогащение	Форматирует строку согласно заданному паттерну			
ВЫРОВНЯТЬ СЛЕВА	обогащение	Выравнивает строку слева. (Добавляет пробелы)			
ВЫРОВНЯТЬ СПРАВА	обогащение	Выравнивает строку справа. (Добавляет пробелы)			
СОЕДИНИТЬ	обогащение	Соединяет строки			
НИЖНИЙ РЕГИСТР	обогащение	Переводит строку в нижний регистр			
ВЕРХНИЙ РЕГИСТР	обогащение	Переводит строку в верхний регистр			
ЗАГОЛОВОК	обогащение	Ставит все первые буквы слов в строке в верхний регистр			
ПОДСТРОКА	обогащение	Извлекает подстроку из данной строки			
ПРОВЕРКА ДЛИНЫ СТРОКИ	валидация	Проверяет строку на минимальное и максимальное количество символов			
МАТЕМАТИЧЕСКИЕ					
ПРОВЕРКА ДИАПАЗОНА ЧИСЕЛ	валидация	Проверяет число на минимальное и максимальное значение			
ПРОВЕРКА ДИАПАЗОНА ДРОБНЫХ ЧИСЕЛ	валидация	Проверяет число на минимальное и максимальное значение			
СУММА	обогащение	Операция сложения			
РАЗНОСТЬ	обогащение	Операция вычитания			
ПРОИЗВЕДЕНИЕ	обогащение	Операция умножения			
ДЕЛЕНИЕ	обогащение	Операция деления			
ОКРУГЛИТЬ	обогащение	Округление числа			
МАКСИМУМ	обогащение	Возвращает максимальное число			
МИНИМУМ	обогащение	Возвращает минимальное число			
ЛОГИЧЕСКИЕ					
AND		Логическое И			
OR		Логическое ИЛИ			
XOR	валидация и	Исключающее ИЛИ			
NOT	обогащение	Инвертирует входное значение(true->false или false->true)			
ISEMPTY		Проверка на отсутствие значения			
РАЗБОР ТИПОВ ДАННЫХ					
ΡΑЗΟБΡΑΤЬ BOOLEAN		Переводит строку в ЛОГИЧЕСКОЕ ЗНАЧЕНИЕ			
РАЗОБРАТЬ DATE		Переводит строку в ДАТА			
РАЗОБРАТЬ NUMBER	обогащение	Переводит строку в ЧИСЛО			
PA3O5PATЬ INTEGER		Переводит строку в ЦЕЛОЕ ЧИСЛО			

НАЗВАНИЕ	тип Функции	ОПИСАНИЕ			
РАЗНЫЕ ЗНАЧЕНИЯ					
ПРОВЕРКА ЗНАЧЕНИЙ	валидация	Проверка значений по регулярному выражению. В качестве входных параметров принимает регулярное выражение и значение для проверки. Значение может быть строковым или числовым			