



CRM-ПЛАТФОРМА «КОНТУР»

Версия системы: 5.1

Руководство по эксплуатации

Программный продукт, описанный в настоящем Руководстве, поставляется строго по лицензионному соглашению. Авторские права ООО «Интерсофт Лаб» защищены законом. Копирование и распространение программного продукта и документации к нему в какой бы то ни было форме и любыми средствами, включая фотокопирование и запись на магнитные носители, в отсутствие специального соглашения является противозаконным и преследуется в судебном порядке.

Компания «Интерсофт Лаб» оставляет за собой право внесения изменений и дополнений в содержание данного Руководства без предварительного уведомления частных лиц и организаций. Компания не несет ответственности за использование настоящего документа и не дает гарантий его коммерческих преимуществ или пригодности для конкретных целей.

© Intersoft Lab 1999-2023. Все права защищены.

Оглавление

Введение	5
Термины и определения.....	6
1 Начало использования Платформы	7
2 Управление репозиторием метаданных.....	7
2.1 Создание метаобъекта	7
2.1.1 Импорт метаописания объекта	8
2.1.2 Пример импорта.....	12
2.2 Выгрузка метаданных	14
3 Импорт данных	14
3.1 Структуры служебных таблиц модуля импорта данных	15
3.1.1 Справочник внешних учетных систем	15
3.1.2 Справочник прикладных групп загрузки данных.....	15
3.1.3 Этапы процесса импорта данных	15
3.1.4 Настройка порядка импорта данных	17
3.1.5 Настройки правил обработки и контроля качества данных....	18
3.2 Структуры таблиц журналов подсистемы импорта данных.....	19
3.2.1 Журнал сеансов импорта данных.....	19
3.2.2 Журнал импортируемых метаобъектов банков данных.....	20
3.2.3 Журнал регистрации исполнения этапов процесса импорта	22
3.2.4 Журнал обработки и контроля качества данных	22
3.2.5 Журнал протокола контроля качества данных	23
3.2.6 Журнал регистрации значений параметров.....	24
3.2.7 Текущее состояние импорта данных.....	24
3.3 Процедуры API Contour ETL.....	25
3.3.1 Процедуры управления процессом	25
3.4 Утилитные процедуры и функции API Contour ETL	31
3.5 Правила использования пользовательских этапов обработки и контроля данных.....	33
3.5.1 Требования к прикладным процедурам, используемым в пользовательских этапах.....	33
3.5.2 Требования к запросу контроля качества данных для процедуры ETL_MASS_ERROR_REGISTER.....	34
3.5.3 Пакетные переменные, разрешенные к использованию в прикладных процедурах	34

3.5.4	Рекомендации к SQL-именам процедур.....	36
3.6	Основной сценарий использования.....	37
3.6.1	Схема исполнения сеанса импорта данных	37
3.6.2	Описание процесса взаимодействия систем в рамках сеанса 38	
	Приложение 1. Пример xml-описания метаобъекта	40
	Приложение 2. XSD-схема словаря метаобъектов ХД	43
	Приложение 3. Пример сценария создания справочников	74
	Приложение 4. Пример сложного сценария импорта метаданных.....	76

Введение

CRM-платформа «Контур» (далее Платформа) является специализированной программной платформой, обеспечивающей построение CRM-систем для финансово-кредитных организаций на базе корпоративного хранилища данных (далее ХД).

Платформа предназначена для быстрого развертывания ХД и прикладных решений на его основе и включает ряд базовых компонент, выполняющих следующие функции:

- Управление метаданными и метаобъектами,
- Управление ETL-процессом.

В данном документе приводится описание базовых функций Платформы, формата и возможных вариантов команд, с помощью которых пользователь выполняет операции с метаданными и данными ХД.

Связанные документы:

- «Программная платформа «Контур». Информация о системе»;
- «Программная платформа «Контур». Инструкция по установке».

Термины и определения

СРМ\СРМ - платформа «Контур» (Платформа,)	Единая инструментально-технологическая среда, предназначенная для построения хранилищ данных и прикладных решений для риск-ориентированного управления бизнесом финансовой организации.
СРМ\СРМ-Приложение	Программные компоненты, обеспечивающих решение отдельных категорий бизнес-задач и путем использования данных, накопленных в Хранилище данных «Контур».
ETL (Extraction, Transformation, Load)	Технология загрузки данных в хранилище данных, включающая в себя последовательные этапы извлечения данных из внешних источников, очистку и преобразование данных в соответствии с требованиями бизнес модели, загрузку в хранилище данных

1 Начало использования Платформы

После выполнения всех действий по настройке СУБД и установке приложения (см. документ «СРМ-платформа «Контур». Инструкция по установке») на сервере будет развернут экземпляр ХД и установлены следующие компоненты:

- Системное приложение – обеспечивает основные механизмы Платформы по манипулированию метаобъектами и данными метаобъектов.
- Базовое приложение – прикладное решение, содержащее настройки stage-области, настройки, необходимые для управления справочниками Платформы, предзаполненные общесистемные справочники, структуры метаобъектов, необходимые для описания организационно-штатной структуры банка.
- Модуль импорта данных – обеспечивает двухфазную загрузку данных из внешних источников, выполнение системных проверок качества данных, подключение любых пользовательских процедур проверки и обработки/обогащения данных на этапах загрузки.
- Модуль обмена метаданными – обеспечивает импорт и экспорт xml-метаописания ХД или его части на основании правил декларативного описания метаобъектов.
- Модуль журнализации – обеспечивает протоколирование действий по управлению данными и метаданными.
- Модуль информационной безопасности – структуры и процедуры необходимые для регистрации и аутентификации пользователей, управления логическими правами пользователей.
- Приложение модели данных «Главная книга» - приложение, формирующее структуры для хранения данных бухгалтерского учета.

Для доступа к данным платформы может быть использовано любой штатный BI-инструмент.

2 Управление метаданными репозиторием

В данном разделе представлено описание функциональности Подсистемы управления метаданными и метаобъектами. Подробная информация о принципах описания метаданных, категориях объектов хранения и структуре объектов представлена в документе «СРМ-платформа «Контур». Информация о системе».

2.1 Создание метаобъекта

Одним из способов создания прикладного метаобъекта в Хранилище данных является импорт его метаописания. Загрузка выполняется модулем обмена метаданными, входящего в состав подсистемы управления метаданными и метаобъектами.

В ходе импорта:

- Проверяется корректность импортируемого описания на соответствие правилам описания структуры прикладных метаобъектов. В случае обнаружения ошибок импорт прерывается.
- Сохраняется метаописание в словаре метаданных;
- Автоматически генерируются таблицы, процедуры и функции для подсистем управления данными и управления импортом данных. В частности, в ХД генерируются:
 - Базовые, центральные, групповые, типовые таблицы;
 - Иерархии,

- Навигаторы;
- Аналитические выборки;
- Stage-таблицы;
- Процедуры для отбора, ввода, удаления и изменения данных.

2.1.1 Импорт метаописания объекта

Для загрузки метаданных в ХД используется команда

isdbinst -m isdbinst.map_walker *ИмяЗагружаемогоФайла* **-log-level**
УровеньЖурнализации **-log-file** *ИмяФайлаЖурнала* **--connection-string**
СтрокаСоединения **--user** *ИмяПользователя* **--password** *Пароль*

Описание ключей загрузки

Ключ	Значение
-m isdbinst.map_walker	Полный путь к загружаемому файлу. Загружаемый файл может быть отдельным XML файлом, содержащим метаинформацию, либо kdl файлом со сценарием загрузки файлов (см. раздел .2.1.1.1).
-log-level	Уровень журнализации. Допустимые значения: 1 – отладочный журнал, 10 подробный журнал, 20 – краткий журнал, 40 – без журнала
-log-file	Имя файла журнала – имя текстового файла, в который будет записан журнал.
--connection-string	Строка соединения с базой банных. Имеет следующий формат. host= Хост port= Порт dbname= ИмяБД options='-csearch_path= Схема '

Импорт должен производиться владельцем схемы.

2.1.1.13 Загрузка из файла

В качестве загружаемого файла может использоваться xml-файл с описанием структуры прикладного метаобъекта. Пример xml-файла для загрузки метаобъекта представлен в Приложении 1. Фрагмент структуры xsd-схемы для описания словаря метаобъектов показан на рис 1., xsd-схема приведена в Приложении 2.

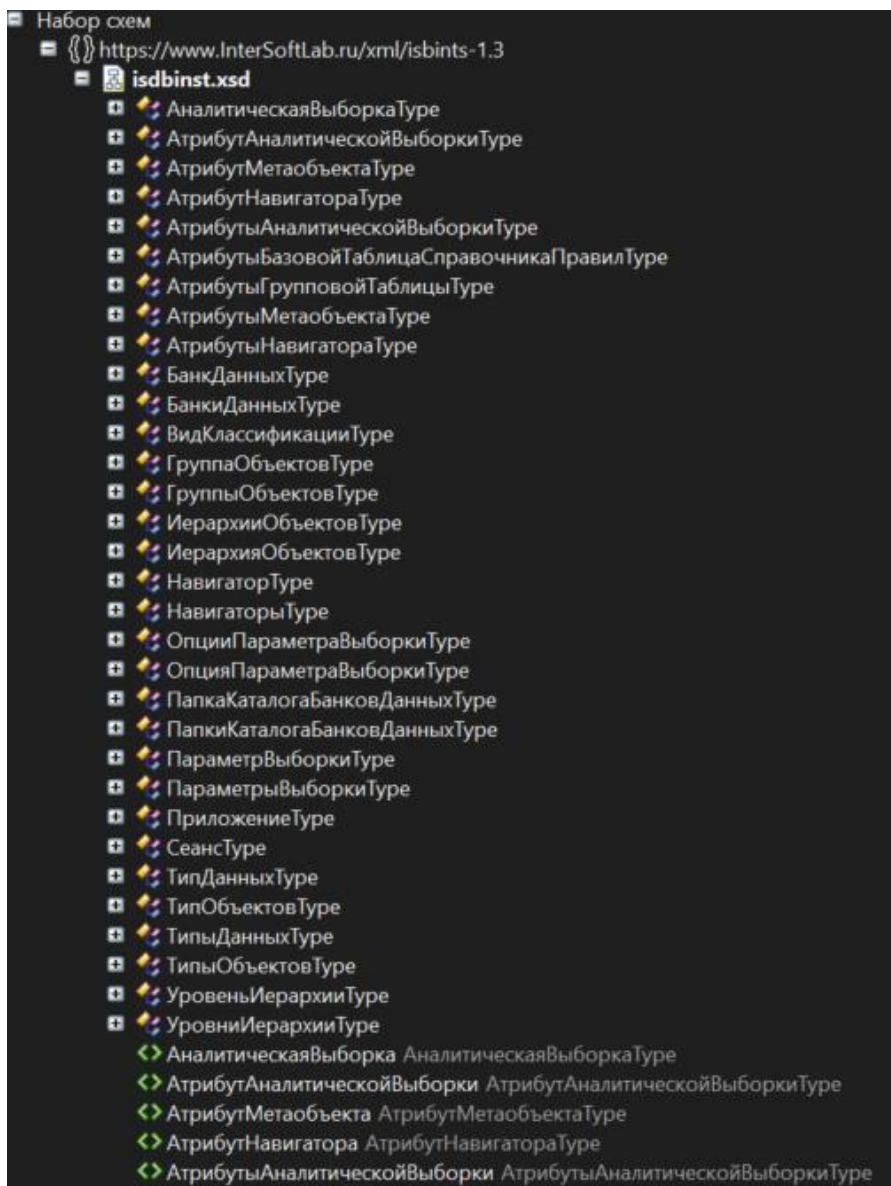


Рис. 1. Структура xsd-схемы

Подсистема «Управление метаданными и метаобъектами» проверяет импортированные метаописания ХД на соответствие xsd-схеме и сохраняет их в словарях системы, а также производит генерацию таблиц, процедур и функций для подсистем управления данными и управления импортом данных.

2.1.1.2 Сценарии загрузки

В качестве загружаемого файла может использоваться kdl файл со сценарием загрузки файлов.

Описание структуры сценария загрузки приводится ниже.

Мар-файлы

```

...

isdbinst version="1.0"
[
исполнительный-узел

```

```

[
исполнительный-узел
]
]
...

## Исполнительные узлы

### log
...

log [level="INFO|DEBUG|WARNING|ERROR|1..40"] "Сообщение"
...

### abort
...

abort "Сообщение"
...

### map
...

map "<путь к мар-файлу относительно данного мар-файла>"
...

### sql
...

sql "<путь к sql-файлу относительно данного мар-файла>"
...

### xml
...

xml "<путь к xml-файлу относительно данного мар-файла>"
...

### data
...

data [p_null=true|false] [p_xml=true|false] "<путь к data.kdl-файлу относительно данного мар-файла>"
...

### layoutxml
...

layoutxml "<путь к xml-файлу относительно данного мар-файла>"

```

```
'''  
  
### if-elif-else  
'''  
  
if {  
    <условный узел>  
    [  
        <исполнительный узел>  
        [  
            <исполнительный узел>  
        ]  
    ]  
}  
[  
    elif {  
        <условный узел>  
        [  
            <исполнительный узел>  
            [  
                <исполнительный узел>  
            ]  
        ]  
    }  
    [  
        elif {  
            <условный узел>  
            [  
                <исполнительный узел>  
                [  
                    <исполнительный узел>  
                ]  
            ]  
        }  
    ]  
    ]  
    [  
        else {  
            [  
                <исполнительный узел>  
                [  
                    <исполнительный узел>  
                ]  
            ]  
        }  
    ]  
]
```

```

}
]
...

## Условные узлы

### bool-query
...

bool-query "SQL-запрос, возвращающий запись с единственным полем типа boolean"
...

```

В Приложении 3 приводятся пример простого сценария импорта (импорт справочников базового приложения Платформы).

В Приложении 4 приводится пример сложного сценария импорта.

2.1.2 Пример импорта

Импорт метаобъекта банк данных «Кредиты ЮЛ». Импорт осуществляется из файла loan.xml (Приложение 1).

Команда загрузки:

```
isdbinst -m isdbinst.map_walker loanj.xml --log-level 10 --log-file isdbinstlog.txt --log-level 20 --connection-string "host=127.0.0.1 dbname=intersoftlab options='-csearch_path=inst,public'" --user dwh --password *****
```

Полученный журнал загрузки (подробный уровень журнализации) имеет вид:

```

Инсталлятор пакетов БД КХД "Контур" 1.1.5 www.InterSoftLab.ru (c) 1999-2023
Установка инициализирована
Рассматривается файл loanj.xml
Обрабатывается файл loanj.xml
Подключение к базе данных dbname=intersoftlab host=127.0.0.1 options=-csearch_path=inst,public user=dwh
Получен объект ПапкаКаталогаБанковДанных( name = ФХД для банков РФ, subsys = C )
обновлен объект ПапкаКаталогаБанковДанных( name = ФХД для банков РФ, subsys = C )
Получен объект ПапкаКаталогаБанковДанных( name = Бухгалтерский учет КО, subsys = C )
обновлен объект ПапкаКаталогаБанковДанных( name = Бухгалтерский учет КО, subsys = C )
Получен объект ПапкаКаталогаБанковДанных( name = Аналитический учет, subsys = C )
обновлен объект ПапкаКаталогаБанковДанных( name = Аналитический учет, subsys = C )
Получен объект ПапкаКаталогаБанковДанных( name = Кредиты физических лиц, subsys = C )
обновлен объект ПапкаКаталогаБанковДанных( name = Кредиты физических лиц, subsys = C )
Получен объект БанкДанных( code = КредитыЮЛ, sys_app = UserBanks, name = Кредиты ЮЛ, note = None, sql_name = DL_LOANJ, structure = None, history = Y, stage = Y, id_type = I, codesize = 50, force_del = None, protect = N )
обновлен объект БанкДанных( code = КредитыЮЛ, sys_app = UserBanks, name = Кредиты ЮЛ, note = None, sql_name = DL_LOANJ, structure = None, history = Y, stage = Y, id_type = I, codesize = 50, force_del = None, protect = N )

```

обновлен объект АтрибутМетаобъекта(category = КредитыЮЛ, code = НомерДоговора, name = Номер договора, note = Номер кредитного договора., field_name = CONTRACT_NO, metaobject = КредитыЮЛ, required = Y, mark = U, ri = N, ref_catg = None, domain = udf_da_code, typesize = None, default = None, protect = N)

Получен объект Метаобъект(category = КредитыЮЛ, sys_app = UserBanks, code = КредитыЮЛ, name = Кредиты ЮЛ, note = None, table_name = DL_LOANJ, kind = C, history = Y, stage = Y, protect = N)

Получен объект БанкДанных(code = ENM_303_LOAN_KIND, sys_app = None, name = None, note = None, sql_name = None, structure = None, history = None, stage = None, id_type = None, codesize = None, force_del = None, protect = None)

Получен объект БанкДанных(code = ENM_303_LOAN_KIND, sys_app = Dictionary, name = Вид ссуды Ф303, note = None, sql_name = ENM_303_LOAN_KIND, structure = None, history = N, stage = N, id_type = C, codesize = 10, force_del = None, protect = N)

Получен объект БанкДанных(code = СпрВидОбработки, sys_app = None, name = None, note = None, sql_name = None, structure = None, history = None, stage = None, id_type = None, codesize = None, force_del = None, protect = None)

Получен объект БанкДанных(code = СпрВидОбработки, sys_app = Dictionary, name = Вид обработки, note = Справочник видов обработки. Определяет перечень возможных видов обработки данных.Поставляется заполненным. В процессе эксплуатации не подлежит изменению., sql_name = ENM_PROCESSING_KIND, structure = None, history = N, stage = N, id_type = C, codesize = 1, force_del = None, protect = N)

Получен объект ГруппаОбъектов(category = КредитыЮЛ, sys_app = UserBanks, code = КредитыЮЛФ303, name = Классификация кредитов ЮЛ Ф303, note = None, table_name = DL_LOANJ_G_F303, history = Y, stage = N, protect = N)

обновлен объект АтрибутМетаобъекта(category = КредитыЮЛ, code = LOAN_KIND, name = Вид ссуды ф.303, note = None, field_name = LOAN_KIND, metaobject = КредитыЮЛФ303, required = Y, mark = U, ri = N, ref_catg = ENM_303_LOAN_KIND, domain = udf_da_code, typesize = 10, default = None, protect = N)

обновлен объект АтрибутМетаобъекта(category = КредитыЮЛ, code = КодКлассиф, name = Код классификации, note = None, field_name = CLS_CODE, metaobject = КредитыЮЛФ303, required = N, mark = U, ri = N, ref_catg = None, domain = udf_da_objcode, typesize = None, default = None, protect = N)

обновлен объект АтрибутМетаобъекта(category = КредитыЮЛ, code = ПравилоКлассиф, name = Правило классификации, note = None, field_name = CLS_RULE_ID, metaobject = КредитыЮЛФ303, required = N, mark = U, ri = N, ref_catg = None, domain = id_int, typesize = None, default = None, protect = N)

обновлен объект АтрибутМетаобъекта(category = КредитыЮЛ, code = ПризнакКлассиф, name = Признак классификации, note = None, field_name = CLS_PROCESSING_KIND, metaobject = КредитыЮЛФ303, required = Y, mark = U, ri = N, ref_catg = СпрВидОбработки, domain = udf_da_code, typesize = 1, default = None, protect = N)

Получен объект Метаобъект(category = КредитыЮЛ, sys_app = UserBanks, code = КредитыЮЛФ303, name = Классификация кредитов ЮЛ Ф303, note = None, table_name = DL_LOANJ_G_F303, kind = G, history = Y, stage = N, protect = N)

обновлен объект ГруппаОбъектов(category = КредитыЮЛ, sys_app = UserBanks, code = КредитыЮЛФ303, name = Классификация кредитов ЮЛ Ф303, note = None, table_name = DL_LOANJ_G_F303, history = Y, stage = N, protect = N)

Установка завершена

2.2 Выгрузка метаданных

Сериализация метаданных любой части ХД в xml может быть использована для реализации следующих сценариев:

- Создания новых прикладных приложений Платформы.
- Хранения версий словарей метаданных.
- Обмена словарями метаданных между разными экземплярами ХД.

Для выгрузки метаданных существующей в ХД категории используется команда:

isdbinst -m isdbinst.tools.dump --directory *ИмяДиректории* --databank *Категории* --log-level *УровеньЖурнализации* --log-file *ИмяФайлаЖурнала* --connection-string *СтрокаСоединения*

Описание ключей выгрузки

Ключ	Значение
--directory	Имя директории для выгрузки
--databank	Имя категории ХД, для которой выгружается информация. Можно казать несколько категорий.
--log-level	Уровень журнализации. Допустимые значения: 1 – отладочный журнал, 10 – подробный журнал, 20 – краткий журнал, 40 – без журнала
--log-file	Имя файла журнала.
--connection-string	Строка соединения с БД. host= Хост port= Порт dbname= ИмяБД options='-csearch_path=

3 Импорт данных

Подробные сведения об организации процесса импорта данных в ХД, структуре хранения данных, фазах и этапах импорта данных, порядке управления процессом загрузки и протоколирования результата приводятся в документе «СРМ-платформа «Контур». Информация о системе».

В данном разделе представлены сведения, необходимые для реализации пользовательского процесса импорта данных в ХД, предполагающего:

- определение очередности обработки и загрузки банков данных в DWH из Stage-области;
- детальное протоколирование процессов импорта в журналах подсистемы импорта данных;
- разработку прикладных процедур обработки и контроля качества данных и использования их на пользовательских этапах импорта.

3.1 Структуры служебных таблиц модуля импорта данных

3.1.1 Справочник внешних учетных систем

Банк данных: «СпрВнешнУчетСистемы» SO_OLTP_SYSTEMS

Банк данных содержит список внешних учетных систем и предназначается для контроля свойства «Учетная система» сеанса загрузки. Историизируемый.

№	Код (SQL-имя)	NULL	Тип данных	Описание
1.	ИдентОбъекта (ID)	N	NUMBER	Идентификатор внешней учетной системы
2.	КодОбъекта (CODE)	N	TEXT(50)	Код внешней учетной системы
3.	Наименование (NAME)	N	TEXT(255)	Краткое наименование внешней учетной системы
4.	ПолнНаим (FULL_NAME)	Y	TEXT(512)	Полное наименование внешней учетной системы
5.	ВерсияУчСист (VERSION)	Y	TEXT(35)	Версия внешней учетной системы.
6.	ДатаНачалаЭксплуат (START_DATE)	Y	DATE	Дата ввода в эксплуатацию
7.	ДатаЗавершЭксплуат (FINISH_DATE)	Y	DATE	Дата завершения эксплуатации. Если система в эксплуатации, то заполняется значением 31.12.9999
8.	NOTE	Y	TEXT(1024)	Описание внешнего источника данных

Настройка справочника выполняется в процессе внедрения и эксплуатации Contour Data Warehouse администратором ХД с помощью специализированного интерфейса.

3.1.2 Справочник прикладных групп загрузки данных

Банк данных: STG_DATABANK_GROUPS

Банк данных содержит список прикладных групп загрузки данных и предназначается для группировки прикладных банков данных Системы при анализе журналов загрузки. Неисториизируемый.

№	Код (SQL-имя)	NULL	Тип данных	Описание
1.	ИдентОбъекта (ID)	N	NUMBER	Идентификатор прикладной группы загрузки данных. Формируется автоматически.
2.	КодОбъекта (CODE)	N	TEXT(25)	Код прикладной группы загрузки данных.
3.	Наименование (DG_NAME)	N	TEXT(255)	Наименование прикладной группы загрузки данных.
4.	Примечание (DG_NOTE)	Y	TEXT(1024)	Примечание к прикладной группе загрузки данных для дополнительного определения ее назначения.
5.	ПредопредПрилож (DG_PREDEFINED)	Y	TEXT(1)	Логический признак, определяющий, что информационная группа предопределена в рамках соответствующего приложения. Возможные значения: 'Y N'

Настройка справочника и определение принадлежности прикладных банков данных к той или иной группе загрузки выполняется администратором ХД в процессе эксплуатации Contour Data Warehouse с помощью специализированного интерфейса «Прикладные группы загрузки данных».

3.1.3 Этапы процесса импорта данных

Имя таблицы: STG_PHASE_SETTINGS

Таблица содержит список этапов процесса обработки данных и предназначается для контроля момента выполнения процедуры обработки данных.

№	SQL-имя	PK	Тип данных	Описание
1.	PH_CODE	✓	TEXT(35)	Уникальный код этапа процесса импорта данных

2.	PH_PARENT	TEXT(35)	Код родительского этапа процесса импорта данных
3.	PH_NAME	TEXT(255)	Наименование этапа процесса импорта данных
4.	PH_ORDER	NUMBER	Порядок выполнения этапов процесса импорта данных
5.	PH_SYSTEM_ONLY	TEXT(1)	Признак, является этап системным или пользовательским. Возможные значения: «Y N». Для пользовательского этапа разрешается регистрация пользовательских процедур обработки и контроля в настройках правил обработки данных Staging Area.
6.	PH_PROCNAME	TEXT(64)	SQL-имя процедуры, отвечающей выполнение этапа
7.	PH_NOTE	TEXT(2000)	Описание назначения этапа

Допустимые значения таблицы:

PH_CODE	PH_PARENT	PH_NAME	PH_ORDER	PH_SYSTEM_ONLY	PH_NOTE
SESSION	NULL	Сеанс импорта данных	0	Y	Регистрация сеанса импорта данных.
STG_LOAD	SESSION	Загрузка данных в Staging Area	1	Y	Этап определяется для определения статистики загрузки данных в Staging Area
ENROLL_PARAMS	STG_LOAD	Регистрация параметров прикладных процедур	2	Y	Регистрация параметров прикладных процедур контроля качества и обработки данных
ENROLL_IMPOBJ	STG_LOAD	Регистрация импортируемых объектов	3	Y	Регистрация импортируемых объектов.
CHECK_IMPOBJ	STG_LOAD	Проверки достаточности структур и взаимосвязей импортируемых объектов	4	Y	Основным назначением этапа является выполнение определенных логических проверок, которые разрешают загрузку данных объектов текущего сеанса.
LOADOBJ_TO_STAGE	STG_LOAD	Загрузка данных в stage-таблицу	5	Y	Загрузка данных в stage-таблицу
DWH_LOAD	SESSION	Старт процесса обработки и загрузки в DWH	10	Y	Старт процесса обработки и загрузки данных в DWH из Staging Area в рамках сеанса.
GLOBAL_CHECK	DWH_LOAD	Общая предварительная обработка данных	20	N	Выполнение пользовательских действий в области Stage до системной проверки для фазы GLOBAL_SYSCHK в рамках сеанса импорта данных.
GLOBAL_SYSCHK	DWH_LOAD	Глобальная системная проверка	30	Y	Общие системные проверки
GLOBAL_BEFORE	DWH_LOAD	Общая дозагрузочная обработка	35	N	Выполнение пользовательских действий до загрузки данных в DWH для фазы BEFORE_LOAD в рамках сеанса импорта данных
START_LOAD_DATA	DWH_LOAD	Подготовка к загрузке данных в DWH	40	Y	Подготовка к загрузке данных из области Stage в область хранилища данных.
DATABANK_BEFORE	START_LOAD_DATA	Дозагрузочная обработка для	41	N	Дозагрузочная обработка для определенного банка данных

		определенного банка данных			
DATABANK_SYS CHK	START_LOA D_DATA	Системные проверки для определенного банка данных	43	Y	Системные проверки для определенного банка данных, в т.ч. проверки значений на NULL и корректности определения рангов для таблиц иерархий. +Дубликаты, +Дата актуальности <> «бесконечность», +Неоднозначность типов
DATABANK_LOA D	START_LOA D_DATA	Загрузка данных отдельного банка данных в DWH	45	Y	Загрузка данных отдельного банка данных в DWH
DATABANK _AFTER	START_LOA D_DATA	Постзагрузочна я обработка для определенного банка данных	47	N	Постзагрузочная обработка для определенного банка данных
GLOBAL_AFTER	DWH_LOAD	Общая постзагрузочна я обработка	50	N	Выполнение постзагрузочных обработок в DWH
END_LOAD_DATA	DWH_LOAD	Завершение загрузки данных в DWH	60	Y	Завершение загрузки данных в DWH, в том числе очистка stage-таблиц и формирование статистического отчета о загрузке данных в DWH

Важно! Этапы DATABANK_BEFORE, DATABANK_LOAD, DATABANK_AFTER выполняются для каждого банка данных, загрузка которого производится в рамках сеанса.

Таблица поставляется заполненной. Изменения состава таблицы производится специалистами системной разработки компании Intersoft Lab.

3.1.4 Настройка порядка импорта данных

Имя таблицы: STG_PROCESS_ORDER

Таблица предназначена для определения очередности обработки и загрузки банков данных из Staging Area в DWH.

№	SQL-имя	PK	NULL	Тип данных	Описание
1.	CO_CATG	✓		TEXT(35)	Код категории. Указатель банка данных. При выборе в форме редактирования отображаются только те банки данных, элементам которых установлен признак OT_STAGE. Ссылка на таблицу DACATOBJ.
2.	ID_ORDER	✓		NUMBER	Порядковый номер выполнения. Определяется очередность банков данных, для которых выполняются процедуры обработки и загрузки.

Пример заполнения таблицы (условно):

Банк данных	Порядок выполнения
Справочник	1
План счетов	2
Орг. структура	3
Клиенты	4
Лицевые счета	5
Проводки	6
Остатки	7
...	...

3.1.5 Настройки правил обработки и контроля качества данных

Имя таблицы: STG_PROCEDURE_SETTINGS

Таблица предназначена для определения порядка выполнения процедур контроля и обработки данных для отдельного банка данных или в общем для сеанса загрузки для этапов, где разрешены пользовательские операции обработки данных.

№	SQL-имя	NUL L	Тип данных	Описание
1.	PS_ID		NUMBER	Идентификатор регистрации процедуры обработки. Автогенерируемый уникальный идентификатор зарегистрированной процедуры.
2.	PS_DATABANK	✓	TEXT(35)	Код банка данных. Указатель банка данных, для которого настраиваются правила обработки данных области Stage.
3.	PS_EXEC_ORDER		NUMBER	Порядковый номер процедуры обработки. Определяет порядок выполнения процедур обработки.
4.	PS_PHASE		TEXT(35)	Этап обработки данных . Ссылка на таблицу Этапы процесса импорта данных
5.	PS_PROCNAME		TEXT(100)	SQL-имя процедуры обработки. Физическое имя процедуры обработки данных.
6.	PS_NOTE	✓	TEXT(500)	Краткое описание назначения процедуры
7.	PS_SYSTEM		TEXT(1)	Признак, определяющий что зарегистрированная процедура является системной и ее регистрация не может быть изменена пользователем в процессе эксплуатации. Возможные значения: Y – системная процедура N – пользовательская процедура Значение по умолчанию: N
8.	PS_EXECUTION_FLAG		TEXT(1)	Флаг, определяющий будет выполняться процедура в рамках автоматического процесса обработки. Возможные значения: Y – исполнять процедуру. N – не исполнять процедуру Значение по умолчанию: Y
9.	PS_CRITICAL_SEVERITY	N	NUMBER(1)	Значение уровня критичности, начиная с которого результат процедуры обработки или контроля считается критической ошибкой и следует прерывание сеанса импорта данных. Допустимые значения: 9
10.	PS_ERROR_SEVERITY	N	NUMBER(1)	Значение уровня критичности, начиная с которого результат процедуры обработки или контроля определен как ошибка, не являющаяся критической. Прерывание сеанса импорта данных не следует. Допустимые значения от 2 до 8
11.	PS_WARNING_SEVERITY	N	NUMBER(1)	Значение уровня критичности, начиная с которого результат процедуры обработки или контроля определен как потенциальная ошибка (предупреждение). Прерывание сеанса импорта данных не следует. Допустимые значения от 1 до 7. Всегда меньше значения поля PS_ERROR_SEVERITY.
12.	PS_CANCEL_ON_EXCEPTION	N	TEXT(1)	Флаг, определяющий выполнять прерывание сеанса при возникновении программного исключения в

<p>процессе исполнения пользовательской процедуры или не прерывать.</p> <p>Возможные значения:</p> <p>Y – прерывать сеанс</p> <p>N – не прерывать сеанс</p> <p>Значение по умолчанию: N</p>

Таблица поставляется заполненной. Изменения состава таблицы производится специалистами прикладной разработки и внедрения компании «Интерсофт Лаб».

Если специалисты Заказчика добавляют собственные процедуры обработки и контроля качества данных при загрузке, то оформление таких процедур должно соответствовать требованиям к оформлению прикладных процедур.

3.2 Структуры таблиц журналов подсистемы импорта данных

3.2.1 Журнал сеансов импорта данных

Имя таблицы журнала: STG_LOG_SESSIONS (вместо ASSCACVT)

Данный журнал предназначен для регистрации события начала сеанса импорта, формирования идентификатора сеанса, а также регистрации события завершения сеанса импорта и фиксации результатов обработки загрузки данных в Staging Area и далее в DWH, в рамках одной записи журнала для каждого сеанса импорта данных.

ColID	ColumnName	PK	AK	Null	DataType	Описание и правила заполнения
1.	SC_ID	✓		N	NUMBER (10)	Идентификатор сеанса импорта. Генерируется автоматически при добавлении новой записи о сеансе импорта данных.
2.	SC_CODE		✓	N	TEXT(128)	Код сеанса импорта.
3.	SC_RUN_COUNT		✓	Y	NUMBER	Количество запусков сеанса импорта данных.
4.	SC_BEGDATE			N	TIMESTAMP	Дата и время начала сеанса импорта.
5.	SC_ENDDATE			Y	TIMESTAMP	Дата и время окончания сеанса импорта.
6.	SC_SESSIONDATE			N	TIMESTAMP	Дата и время окончания сеанса выгрузки данных из внешней учетной системы.
7.	SC_DATABEGIN			N	DATE	Дата начала периода, с которой были выгружены данные из учетной системы в рамках сеанса выгрузки.
8.	SC_DATAEND			N	DATE	Дата, по которую были выгружены данные из учетной системы в рамках сеанса выгрузки.
9.	SC_FILENAME			Y	TEXT (1024)	Если используется сервер обмена CONTOUR, то указывается путь и имя архивного файла. Если используется ETL-инструмент стороннего производителя, то заполняется идентификатор внешнего процесса этого ETL-инструмента.
10.	SC_FLCOUNTER			Y	NUMBER (10)	Количество файлов в архивном файле.
11.	SC_FLERCOUNTER			Y	NUMBER (10)	Количество файлов при загрузке которых произошли ошибки.

12.	SC_CRC	Y	NUMBER (10)	Значение CRC загружаемых данных
13.	SC_SUCCESS	Y	CHAR(1)	Флаг, определяющий успешность загрузки данных в Staging Area. Возможные значения: «Y N W»: ‘Y’ – данные загружены успешно; ‘N’ – не удалось загрузить данные в штатном режиме; ‘W’ – данные загружены, но есть предупреждения внешнего процесса;
14.	SC_ERRMSG	Y	TEXT (4000 Byte)	Сообщение об ошибке в случае неуспешного завершения сеанса. В случае прерывания сеанса по исключению, в данный атрибут записывается сообщение об ошибке и стек вызовов процедур.
15.	SC_OLTP_ID	N	NUMBER(10)	ID внешней учетной системы из которой поступили данные. Ссылка на банк данных «Внешние учетные системы».
16.	SC_BRANCH_ID	N	NUMBER(10)	ID филиала, данные которого загружаются в ХД. Ссылка на банк данных «Организационная структура»
17.	SC_DWH_SUCCESS	Y	TEXT(1)	Статус завершения загрузки данных в DWH. Ссылка на таблицу Виды журнальных сообщений NULL - допустимо только в том случае если сеанс загрузки в ХД завершился не штатно (аварийно).
18.	SC_NOTE	Y	TEXT (1024)	Комментарий или примечание к сеансу загрузки.
19.	SC_EXECUTOR	N	TEXT (35)	Логин пользователя СУБД, в рамках сессии которого выполняется процесс. Заполняется значением системной функции USER .

3.2.2 Журнал импортируемых метаобъектов банков данных

Имя таблицы журнала: STG_LOG_LOADOBS (вместо ASIMPCVT)

Данный журнал предназначен для регистрации загрузки данных категорий и их метаобъектов, полученных в рамках сеанса загрузки данных в Staging Area.

№	Column Name	PK	AK	Null	Data Type	Description
1.	IC_ID	✓		N	NUMBER (10)	Идентификатор импортируемого объекта. Генерируется автоматически при добавлении новой записи о загружаемом объекте в рамках сеанса.
2.	IC_SESSION		✓	N	NUMBER (10)	Идентификатор сеанса импорта данных. Ссылка на таблицу Журнал сеансов импорта данных .
3.	IC_DATABANK		✓	N	TEXT(35)	Код категории. Заполняется значением кода категории, в которую входит импортируемый метаобъект. Физическая ссылка на таблицу DACATOBJ
4.	IC_METAOBJ		✓	N	TEXT(35)	Код метаобъекта, который передается в процедуру регистрации импортируемых объектов.

5.	IC_OBJTYP	✓	Y	NUMBER(10)	ID метаобъекта, данные которого импортируются в рамках сеанса. Физическая ссылка на таблицу DAOBJTYP.
6.	IC_OBJHIER	✓	Y	NUMBER(10)	ID иерархии, данные которой импортируются в рамках сеанса. Физическая ссылка на таблицу DASTROBJ.
7.	IC_STG_START		Y	TIMESTAMP	Дата и время начала импорта данных метаобъекта в Stage Area.
8.	IC_STG_END		Y	TIMESTAMP	Дата и время окончания импорта данных метаобъекта в Stage Area.
9.	IC_STG_TOTALTIME			INTERVAL DAY(2) TO SECOND(6)	Реальная длительность загрузки метаобъекта в Staging Area в рамках сеанса.
10.	IC_DWH_START		Y	TIMESTAMP	Дата и время начала импорта данных метаобъекта в DWH.
11.	IC_DWH_END		Y	TIMESTAMP	Дата и время окончания импорта данных метаобъекта в Stage Area.
12.	IC_DWH_TOTALTIME		Y	INTERVAL DAY(2) TO SECOND(6)	Реальная длительность загрузки метаобъекта в Staging Area в рамках сеанса.
13.	IC_FILENAME		Y	TEXT (1024)	Примечание или комментарий к загружаемому объекту. Может содержать любую информацию, связанную с объектом, например имя файла.
14.	IC_LOADERERR		Y	TEXT(50)	Код возврата утилиты загрузки данных в Staging Area для соответствующего загружаемого объекта. Например, SQL* Loader.
15.	IC_LOADLOG		Y	CLOB	Лог (файл отчета) утилиты загрузки в Staging Area для соответствующего загружаемого объекта
16.	IC_LOADOUT		Y	CLOB	Текст возврата утилиты загрузки в Staging Area для соответствующего загружаемого объекта
17.	IC_STG_SUCCESS		Y	TEXT(1)	Флаг завершения загрузки в Staging Area . Возможные значения: «Y N» NULL - допустимо только в том случае если сеанс загрузки в Staging Area завершился не штатно (аварийно).
18.	IC_REC_STAGE		Y	NUMBER	Количество записей загруженных в соответствующую stage-таблицу для объекта.
19.	IC_DWH_SUCCESS		Y	TEXT(1)	Флаг завершения загрузки в DWH. Ссылка на таблицу Виды журнальных сообщений Возможные значения: R – завершено с критическими ошибками E – завершено с ошибками; W – завершено успешно с предупреждениями; I - завершено успешно с информационными сообщениями; N - завершено успешно NULL - допустимо только в том случае если сеанс загрузки в DWH завершился не штатно (аварийно).
20.	IC_REC_DWH_INS		Y	NUMBER	Количество записей вставленных в соответствующей таблицы DWH для метаобъекта.
21.	IC_REC_DWH_UPD		Y	NUMBER	Количество обновленных записей в соответствующей таблицы DWH для метаобъекта.

22.	IC_REC_DWH_DEL	Y	NUMBER	Количество удаленных записей из соответствующей таблицы DWH для метаобъекта.
-----	----------------	---	--------	--

3.2.3 Журнал регистрации исполнения этапов процесса импорта

Таблица: STG_LOG_PROCESSES (вместо STG_PROCESS_LOG)

Таблица предназначена для регистрации исполнения этапов процесса загрузки данных, произошедших в рамках сеанса загрузки.

№	SQL-имя	PK	AK	NULL	Тип данных	Описание
1.	LP_ID	✓		N	NUMBER	Идентификатор этапа сеанса загрузки для загружаемого объекта. Формируется автоматически при добавлении записи об этапе загрузки для объекта.
2.	LP_SESSION		✓	N	NUMBER	Идентификатор сеанса. Ссылка на таблицу Журнал сеансов импорта .
3.	LP_PHASE		✓	N	TEXT(35)	Идентификатор этапа процесса обработки. Ссылка на таблицу Этапы процесса импорта данных .
4.	LP_LOADOBJ		✓	Y	NUMBER	Идентификатор загружаемого объекта. Ссылка на таблицу Журнал импортируемых объектов . Значение NULL используется для этапов, которые являются общими, а не относятся к циклу импорта данных по метаобъектам.
5.	LP_PROCNAME		✓	N	TEXT(255)	SQL-имя исполняемой процедуры в рамках этапа.
6.	LP_START			N	TIMESTAMP	Дата и время начала этапа обработки
7.	LP_END			Y	TIMESTAMP	Дата и время окончания этапа обработки
8.	LP_TOTALTIME			Y	INTERVAL DAY(2) TO SECOND(6)	Реальная длительность обработки этапа для загружаемого объекта в рамках сеанса.
9.	LP_RESULT			Y	TEXT(1)	Флаг завершения этапа процесса загрузки. Ссылка на таблицу Виды журнальных сообщений
10.	LP_MESSAGE			Y	CLOB	Текст сообщения для этапа обработки

3.2.4 Журнал обработки и контроля качества данных

Имя таблицы: STG_LOG_CHECKS (вместо STG_CHECK_LOG)

Таблица предназначена для регистрации событий и сбора статистики исполнения системных и пользовательских процедур обработки и контроля качества в рамках сеанса загрузки на соответствующем этапе.

№	SQL-имя	PK	AK	NULL	Тип данных	Описание
1.	LC_ID	✓		N	NUMBER	Идентификатор записи журнала обработки и контроля качества данных. Формируется автоматически.
2.	LC_SESSION		✓	N	NUMBER	Идентификатор сеанса загрузки. Ссылка на таблицу «Журнал сеансов загрузки» .
3.	LC_SESSION_PHASE		✓	N	NUMBER	Идентификатор этапа обработки.

					Ссылка на таблицу Журнал регистрации этапов процесса загрузки .
4.	LC_CHK_RULE	✓	N	NUMBER	Идентификатор правила обработки или контроля качества данных. Ссылка на таблицу Настройки правил обработки и контроля качества данных в Staging Area
5.	LC_START		N	TIMESTAMP	Дата и время начала работы процедуры обработки или контроля качества данных.
6.	LC_END		Y	TIMESTAMP	Дата и время окончания процедуры обработки или контроля качества данных.
7.	LC_TOTALTIME		Y	INTERVAL DAY(2) TO SECOND(6)	Длительность работы процедуры обработки или контроля качества данных.
8.	LC_RESULT		Y	TEXT(1)	Флаг возврата работы процедуры обработки. Ссылка на таблицу Виды журнальных сообщений Возможные значения: NULL определяется как аварийное завершение процедуры. 'R' – в процессе исполнения процедуры были обнаружены критические ошибки 'E' – в процессе исполнения процедуры были обнаружены ошибки 'W' - в процессе исполнения процедуры были предупреждения 'I' - в процессе исполнения процедуры были информационные сообщения 'N' - в процессе исполнения процедуры не было сообщений
9.	LC_CRITICAL_QNTY		Y	NUMBER	Количество критических ошибок, обнаруженных в процессе работы процедуры
10.	LC_ERRORS_QNTY		Y	NUMBER	Количество ошибок, обнаруженных в процессе работы процедуры
11.	LC_WARN_QNTY		Y	NUMBER	Количество предупреждений определенных процедурой обработки
12.	LC_PROTOCOL		Y	CLOB	Протокол (отчет) работы процедуры обработки или контроля качества данных

3.2.5 Журнал протокола контроля качества данных

Имя таблицы: STG_LOG_ERRORS

Таблица предназначена для регистрации сообщений об ошибках или предупреждениях, формируемых процедурами обработки данных в рамках процесса контроля качества данных.

Запись в таблицу осуществляется процедурами регистрации сообщений для объекта данных.

№	SQL-имя	PK	NULL	Тип данных	Описание
1.	ER_ID	✓	N	NUMBER	Уникальный идентификатор записи об ошибке. Формируется с помощью автоинкрементации.
2.	ER_STG_BIGINT		Y	BIGINT	BIGINT записи таблицы ETL_S_*.
3.	ER_SESSION		N	NUMBER	Идентификатор сеанса. Ссылка на таблицу STG_LOG_SESSIONS.
4.	ER_DATABANK		Y	TEXT(35)	Код категории. Ссылка на таблицу DACATOBJ.
5.	ER_LOADOBJ		Y	NUMBER	Идентификатор сеанса импорта данных по метаобъекту. Ссылка STG_LOG_LOADOBS
6.	ER_CHECK		N	NUMBER	Идентификатор записи журнала процедур обработки/контроля качества.

7.	ER_DWH_BIGINT	Y	BIGINT	Ссылка на таблицу STG_LOG_CHECKS. BIGINT соответствующей записи соответствующей таблицы КХД По умолчанию NULL
8.	ER_MSG	Y	VARCHAR(2000)	Сообщение ошибки, которую вернула процедура обработки для конкретной записи соответствующей stage-таблицы.
9.	ER_SEVERITY	N	NUMBER(1)	Уровень критичности ошибки По умолчанию 0

3.2.6 Журнал регистрации значений параметров

Имя таблицы: STG_LOG_PARAMS (вместо ASPARCVT)

Таблица предназначена для регистрации значений параметров процедур обработки и контроля качества данных, выполняемых на соответствующем этапе сеанса загрузки для загружаемого объекта или сеанса в целом.

ColID	Column Name	PK	AK	Null	Data Type	Description
1.	PC_ID	✓		N	NUMBER (10)	Идентификатор значения параметра. Формируется автоматически.
2.	PC_SESSION		✓	N	NUMBER (10)	Идентификатор сеанса загрузки. Ссылка на таблицу « Журнал сеансов загрузки ».
3.	PC_PHASE		✓	N	TEXT(35)	Код этапа обработки. Ссылка на таблицу Этапы процесса импорта данных
4.	PC_CHK_RULE		✓	N	NUMBER (10)	Идентификатор зарегистрированной процедуры обработки. Ссылка на таблицу Настройки правил обработки и контроля качества данных
5.	PC_PARAMETER		✓	N	TEXT(30)	SQL-имя параметра процедуры.
6.	PC_DATE			Y	DATE	Атрибут хранения значения для параметра с типом DATE.
7.	PC_NUMBER			Y	NUMBER	Атрибут хранения значения для параметра с типом NUMBER.
8.	PC_TEXT			Y	TEXT(4000)	Атрибут хранения значения для параметра с типом NUMBER.

3.2.7 Текущее состояние импорта данных

Имя таблицы: STG_LOG_CURRENTS

Таблица предназначена для мониторинга текущего состояния загрузки и содержит в себе по одной записи на каждый активный сеанс загрузки. После корректного завершения сеанса запись о нем удаляется из таблицы.

ColID	Column Name	PK	Null	Data Type	Description
1.	CU_SESSION		N	NUMBER (10)	Идентификатор сеанса загрузки. Ссылка на таблицу « Журнал сеансов загрузки ».
2.	CU_SESSION_CODE		N	TEXT(128)	Код сеанса загрузки.
3.	CU_PHASE		N	TEXT(35)	Код текущего этапа исполнения
4.	CU_DATABANK		Y	TEXT(35)	Код категории, данные которой загружаются в текущий момент исполнения
5.	CU_METAOBJ		Y	TEXT(128)	Код метаобъекта, данные которого загружаются в текущий момент исполнения
6.	CU_PROCEDURE		Y	TEXT(128)	SQL-имя текущей процедуры обработки данных
7.	CU_START		N	TIMESTAMP	Дата и время регистрации/обновления записи.
8.	CU_SID		N	NUMBER	Значение SID для текущего сеанса загрузки

3.3 Процедуры API Contour ETL

3.3.1 Процедуры управления процессом

Процедура ETL_SESSION_REGISTER

Процедура предназначена для регистрации сеанса в журнале сеансов импорта данных (таблица STG_LOG_SESSIONS).

В рамках процедуры выполняются:

1. Контроль кода сеанса.
Если в параметр кода сеанса передано значение NULL, то процедура прерывает свою работу по программному исключению.
2. Контроль наличия успешно завершенного сеанса в журнале сеансов.
Если найден сеанс с кодом равным переданному и он завершен успешно, то процедура прерывает свою работу по программному исключению.
3. Регистрация сеанса.
4. Возврат идентификатора сеанса.

Синтаксис

```
ETL_SESSION_REGISTER(p_session, p_session_code, p_databegin, p_dataend, p_filename,
                     p_branch, p_flcount, p_crc, p_sessdate, p_oltp, p_note)
```

Параметры

p_session in out **number(10)**

Параметр возвращает значение идентификатора сеанса, которое автогенерируется при регистрации сеанса. Полученное значение в дальнейшем используется для передачи в другие процедуры API в рамках исполнения внешнего процесса.

p_session_code in **TEXT(50)**

Параметр принимает значение уникального кода сеанса, который формируется внешним процессом ETL. Формат кода определяется в каждом отдельном проекте внедрения.

p_databegin in **date** default null

Параметр принимает значение даты начала периода, за который выгружены данные из учетной системы.

p_dataend in **date** default null

Параметр принимает значение даты завершения периода, за который выгружены данные из учетной системы.

p_filename in **TEXT(1024)** default null

Параметр принимает значение:

- При использовании сервера файлового обмена Contour - путь и имя заархивированного файла выгруженных данных
- При использовании стороннего инструмента ETL – код сеанса процесса ETL, исполняемого в рамках такого инструмента

p_branch in **TEXT(50)** default null

Параметр принимает значение кода филиала, данные которого будут загружаться в рамках сеанса. Код филиала должен соответствовать одному из значений кодов филиалов, определенных в банке данных «Организационная структура».

p_flcount in **number(10)** default null

Параметр принимает значение:

- При использовании сервера файлового обмена Contour – количество файлов в заархивированном файле выгруженных данных
- При использовании стороннего инструмента ETL – количество метаобъектов предполагаемых к загрузке в процессе сеанса ETL, исполняемого в рамках такого инструмента

p_crc in **number(10)** default null

Параметр принимает значение контрольной суммы (CRC) заархивированного файла выгруженных данных. Как правило, заполняется при использовании сервера файлового обмена Contour

p_sessdate in **date** default null

Параметр принимает значение даты и времени окончания выгрузки данных из учетной системы

p_oltp in **TEXT(50)** default null

Параметр принимает значение кода внешней учетной системы, данные которой будут загружаться в рамках сеанса. Указываемый код внешней учетной системы должен соответствовать одному из значений кодов внешних учетных систем, определенных в банке данных «Внешние учетные системы».

p_note in **TEXT(1024)** default null

Параметр принимает значение текста комментария к сеансу загрузки.

Пример вызова

3.3.1.1.1

```
ETL_session_register(p_session,
                        <код сеанса>,
                        <дата начала данных>,
                        <дата окончания данных>,
                        <имя файла|код сеанса внешнего ETL>,
                        <код филиала>,
                        <количество файлов|метаобъектов>,
                        <контрольная сумма>,
                        <дата и время окончания выгрузки>,
                        <код внешней учетной системы>,
                        <комментарий к сеансу загрузки>);
```

Процедура ETL_LOADOBJ_REGISTER

Процедура предназначена для регистрации метаобъекта, данные которого предполагается загружать в рамках сеанса загрузки, в журнале импорта данных по метаобъектам (таблица STG_LOG_LOADOBS).

В рамках процедуры выполняются:

1. Контроль параметров.
Если в параметры ИД сеанса или код банка данных или код метаобъекта передано значение NULL, то процедура прерывает свою работу по программному исключению.
2. Контроль наличия зарегистрированного сеанса в журнале сеансов.
Если не найден сеанс с ИД равным переданному, то процедура прерывает свою работу по программному исключению.
3. Контроль наличия банка данных в метаданных системы по параметру кода банка данных.
Если код банка данных не найден, то процедура прерывает свою работу по программному исключению.
4. Регистрация метаобъекта в журнале импорта данных по метаобъектам.
5. Инициализация Staging Area для метаобъекта
6. Возврат идентификатора регистрации метаобъекта.

Синтаксис

ETL_LOADOBJ_REGISTER(*p_session*, *p_databank*, *p_metaobj*, *p_filename*, *p_note*, *p_loadobj*)

Параметры

p_session in **number(10)**

Параметр принимает значение идентификатора сеанса, в рамках которого производится импорт данных.

p_databank in **TEXT(35)**

Параметр принимает значение кода банка данных, которому принадлежит регистрируемый метобъект.

p_metaobj in **TEXT(35)**

Параметр принимает значение кода регистрируемого метаобъекта.

p_filename in **TEXT(1024)**

Параметр принимает значение:

- При использовании сервера файлового обмена Contour – имя файла загружаемых в метаобъект данных
- При использовании стороннего инструмента ETL – дополнительная информация по метобъекту, формируемая в рамках внешнего процесса.

p_note in **TEXT(1024)** default null

Параметр принимает значение текста комментария к загрузке данных по метаобъекту.

p_loadobj in out **number(10)**

Параметр возвращает autogenerated значение идентификатора регистрации метаобъекта, для которого будет производиться импорт данных в рамках сеанса. Полученное значение в дальнейшем может быть использовано для передачи в другие процедуры API в рамках исполнения внешнего процесса.

Пример вызова

```
ETL_loadaobj_register(<ИД сеанса>,  
                        <код банка данных>,  
                        <код метаобъекта>,  
                        <имя файла данных>  
                        <комментарий к загрузке данных>,  
                        p_loadobj);
```

Процедура ETL_LOADOBJ_STG_LOAD

Процедура предназначена для регистрации начала импорта данных в Staging Area для метаобъекта в журнале процесса импорта данных (таблица STG_LOG_PROCESSES).

В рамках процедуры выполняются:

1. Контроль параметров.
Если в параметр ИД регистрации метаобъекта передано значение NULL, то процедура прерывает свою работу по программному исключению.
2. Контроль наличия регистрации метаобъекта в журнале импортируемых метаобъектов по сеансу.
Если **не** найдена регистрация метаобъекта с ИД равным переданному, то процедура прерывает свою работу по программному исключению.
3. Регистрация начала импорта данных по метаобъекту в журнале процесса импорта данных.
4. Ничего не возвращает

Синтаксис

ETL_LOADOBJ_STG_LOAD(*p_loadobj*)

Параметры

p_loadobj in **number(10)**

Параметр принимает значение идентификатора регистрации метаобъекта по сеансу.

Пример вызова

```
ETL_loadaobj_stg_load(<ИД регистрации метаобъекта>);
```

Процедура ETL_LOADOBJ_STG_FINALIZE

Процедура предназначена для подтверждения импорта данных в Staging Area для метаобъекта в журналах метобъектов по сеансу (таблицы STG_LOG_LOADOBJJS) и процесса импорта данных (таблица STG_LOG_PROCESSES).

В рамках процедуры выполняются:

1. Контроль параметров.
Если в параметр ИД регистрации метаобъекта передано значение NULL, то процедура прерывает свою работу по программному исключению.
2. Контроль наличия регистрации метаобъекта в журнале импортируемых метаобъектов по сеансу.
Если **не** найдена регистрация метаобъекта с ИД равным переданному, то процедура прерывает свою работу по программному исключению.
3. Обновляется запись регистрации загружаемого метаобъекта в STG_LOG_LOADOBJJS
4. Подтверждается исполнение этапа для загружаемого метаобъекта в STG_LOG_PROCESSES
5. Ничего не возвращает

Синтаксис

ETL_LOADOBJ_STG_FINALIZE(*p_loadobj*, *p_success*, *p_loadererr*, *p_loadlog*, *p_loadout*)

Параметры

p_loadobj in **number(10)**

Параметр принимает значение идентификатора регистрации метаобъекта по сеансу.

p_success in **TEXT(1)** default 'Y'

Параметр принимает значение статуса импорта данных в Staging Area для зарегистрированного метаобъекта:

Возможные значения:

- 'Y' – загрузка в Staging Area выполнена успешно
- 'N' – загрузка в Staging Area завершилась неуспешно

p_loadererr in **TEXT(15)** default null

Параметр принимает значение кода возврата утилиты загрузки SQL*Loader либо внешнего инструмента ETL.

p_loadlog in **clob** default null

Параметр принимает как значение содержание протокола исполнения загрузки утилиты загрузки SQL*Loader либо внешнего инструмента ETL.

p_loadout in **clob** default null

Параметр принимает как значение содержание файла вывода внешнего инструмента ETL.

Пример вызова

```
ETL_loadobj_stg_finalize(<ИД регистрации метаобъекта>,
                        <статус импорта данных>,
                        <код возврата инструмента загрузки>,
                        <содержание протокола загрузки>,
                        <содержание файла вывода>);
```

Процедура ETL_SESSION_STG_FINALIZE

Процедура предназначена для подтверждения завершения сеанса импорта данных в Staging Area. В результате обновляется запись журнала сеансов (таблица STG_LOG_SESSIONS) в части атрибутов SC_SUCCESS, SC_FLERCOUNT и SC_ERRMSG.

В рамках процедуры выполняются:

1. Контроль заполнения параметра статуса импорта.
Если параметр статуса импорта равен NULL, то процедура прерывает свою работу по программному исключению.
2. Обновление записи по сеансу.

3. Процедура ничего не возвращает.

Синтаксис

ETL_SESSION_STG_FINALIZE(*p_stg_success*, *p_stg_flercount*, *p_stg_errmsg*)

Параметры

p_stg_success in **number(10)**

Параметр принимает значение статуса завершения сеанса импорта данных в Staging Area.

Возможные значения:

- 'Y' – загрузка данных выполнена успешно
- 'N' – загрузка данных завершилась **неуспешно**

p_stg_flercount in **number(10) default NULL**

Параметр принимает значение количества метаобъектов, при загрузке данных которых в Staging Area были обнаружены ошибки

p_errmsg in **TEXT(4000) default NULL**

Параметр принимает как значение содержание сообщения переданного из внешнего процесса, как комментарий к завершению сеанса, в т.ч. дополнительной информации об ошибках.

Пример вызова

```
ETL_session_stg_finalize(<статус завершения сеанса>,
                        <количество метаобъектов с ошибками>,
                        <сообщение внешнего процесса>);
```

Процедура ETL_PARAMVALUE_REGISTER

Процедура предназначена для регистрации значений параметров прикладных процедуры обработки и контроля качества данных, которые исполняются в рамках сеанса загрузки. Регистрация значений выполняется в таблицу STG_LOG_PARAMS.

В рамках процедуры выполняются:

1. Контроль параметров:
 - a. Если P_PHASE или P_PROC или P_NAME равны NULL, то формируется программное исключение с соответствующим сообщением.
 - b. Если P_DATABANK равен NULL и P_PHASE равно ('BEFORE_DATABANK','AFTER_DATABANK') , то формируется программное исключение с соответствующим сообщением.
2. Контроль исполнения:

Если найдена регистрация для значения в таблице STG_LOG_PARAMS, то формируется программное исключение с соответствующим сообщением.
3. Выполняется приведение типа данных для значения P_VALUE на основании метаинформации СУБД. Если приведение не удалось выполнить, то формируется программное исключение с соответствующим сообщением.
4. Регистрируется значение параметра в таблице STG_LOG_PARAMS.
5. Ничего не возвращается.

Синтаксис

3.3.1.1.2 ETL_PARAMVALUE_REGISTER(*p_phase*, *p_databank*, *p_proc*, *p_name*, *p_value*)

Параметры

p_phase in **TEXT(35)**

Параметр принимает значение этапа сеанса импорта данных, в рамках которого исполняется прикладная процедура.

p_databank in **TEXT(35) default NULL**

Параметр принимает значение кода банка данных, для которого зарегистрирована процедура P_PROC на этапах P_PHASE IN ('BEFORE_DATABANK','AFTER_DATABANK')

p_proc in **TEXT(30)** default NULL

Параметр принимает SQL-имя процедуры, для которой регистрируется значение параметра.

p_name in **TEXT(30)** default NULL

Параметр принимает SQL-имя параметра процедуры P_PROC, для которого регистрируется значение параметра.

p_value in **TEXT(30)** default NULL

Параметр принимает значение параметра P_NAME процедуры P_PROC.

Пример вызова

```
ETL_paramvalue_register(<этап процесса импорта данных>,
                        <код банка данных>,
                        <SQL-имя процедуры>,
                        <SQL-имя параметра>,
                        <значение параметра>);
```

Процедура ETL_PROCESS_SESSION

Процедура предназначена для запуска процесса переноса данных из Staging Area в Хранилище данных..

В рамках процедуры выполняются:

1. Контроль наличия уже запущенного сеанса переноса данных в ХД.
Если в текущих активных сессиях к СУБД обнаружена запись с текущим ИД сеанса, то процедура прерывает свою работу по программному исключению.
2. Исполнение пользовательских процедур контроля данных по сеансу.
3. Исполнение системных процедур контроля данных по сеансу.
4. Исполнение предзагрузочных пользовательских процедур контроля данных по сеансу.
5. Перенос данных в ХД
6. Исполнение постзагрузочных пользовательских процедур контроля данных по сеансу.
7. Очистка области Stage по сеансу.
8. Возвращает статус исполнения переноса данных из Staging Area в Хранилище данных

Синтаксис

```
ETL_PROCESS_SESSION(p_session_load, run_mode, id_phase, co_catg, is_continued,
                    p_check_status, p_dwh_success)
```

Параметры

p_session_load in **number**

Параметр принимает значение идентификатора сеанса, для которого выполняется перенос данных из Staging Area в Хранилище данных.

run_mode in **TEXT** default 'FULL'

Параметр не используется, оставлен для обеспечения совместимости вызова процедуры.

id_phase in **TEXT** default NULL

Параметр не используется, оставлен для обеспечения совместимости вызова процедуры.

co_catg in **TEXT** default NULL

Параметр не используется, оставлен для обеспечения совместимости вызова процедуры.

is_continued in **TEXT** default NULL

Параметр не используется, оставлен для обеспечения совместимости вызова процедуры.

p_check_status in **TEXT** default 'Y'

Параметр не используется, оставлен для обеспечения совместимости вызова процедуры.

p_dwh_success in out **TEXT(1)**

Параметр возвращает значение общего статуса переноса данных из Staging Area в Хранилище данных по сеансу импорта данных.

Возможные значения:

- 'R' – перенос данных завершился с критической ошибкой, сеанс прерван;
- 'E' – перенос данных завершился с ошибками без прерывания сеанса;
- 'W' – перенос данных завершился с предупреждениями без прерывания сеанса;
- 'I' – перенос данных завершился с информационными сообщениями без прерывания сеанса;
- 'N' – перенос данных завершился в нормальном режиме без прерывания сеанса;

Рекомендуемый пример вызова

```
ETL_process_session(<ИД сеанса>,  
                   <статус исполнения переноса данных>);
```

Внимание! Также поддерживается вызов в рамках обратной совместимости версий.

3.4 Утилитные процедуры и функции API Contour ETL

Процедура ETL_ERROR_REGISTER

Процедура предназначена для позаписной регистрации результата контроля качества данных в журнале контроля качества STG_LOG_ERRORS.

В рамках процедуры выполняются:

1. Контроль параметров:
Если P_SEVERITY равен NULL, то формируется программное исключение с соответствующим сообщением.
2. Регистрируется запись в журнале контроля качества STG_LOG_ERRORS.
3. Возвращаются:
 - a. Идентификатор записи журнала контроля качества (ER_ID)
 - b. Результат исполнения процедуры (P_RESULT)
 - c. Сообщение результата исполнения процедуры (P_RESULT_MSG)

Все программные исключения, которые могут возникнуть в ходе выполнения перехватываются и процедура возвращает данные исключения в параметрах исполнения результата.

Синтаксис

```
ETL_ERROR_REGISTER(p_er, p_stg_BIGINT, p_dwh_BIGINT, p_msg, p_severity, p_result,  
                  p_result_msg)
```

Параметры

p_er in out **number** default null

Параметр возвращает идентификатор записи журнала протокола контроля качества данных STG_LOG_ERRORS

p_stg_BIGINT in **BIGINT** default null

Параметр принимает значение BIGINT записи соответствующей stage-таблицы ETL_S... для которой формируется запись в журнала протокола контроля качества данных.

p_dwh_BIGINT in **BIGINT** default null

Параметр принимает значение BIGINT записи соответствующей таблицы метаобъекта Хранилища данных для которой формируется запись в журнала протокола контроля качества данных.

p_msg in **TEXT(2000)** default null

Параметр принимает текст сообщения о результате контроля качества данных для записи в журнал протокола контроля качества данных.

p_severity in number

Параметр принимает значение критичности сообщения для записи в журнал протокола контроля качества данных.

Допустимые значения: [0-9]

- 0 – нормальное сообщение.
- Промежуточные уровни настраиваются для каждой прикладной процедуры.
- 9 – критическое сообщение.

p_result in out number

Параметр возвращает значение кода результата исполнения данной процедуры.

Возможные значения:

- 0 – процедура отработала успешно
- (-1) – процедура отработала не успешно
- (-2) – не задан обязательный параметр уровня критичности ошибки

p_result_msg in out TEXT

Параметр возвращает сообщение результата исполнения данной процедуры.

Рекомендуемый пример вызова

```
ETL_error_register(p_er,
                  <BIGINT записи stage-таблицы>,
                  <BIGINT записи таблицы метаобъекта ХД>,
                  <сообщение о результате контроля>,
                  <уровень критичности результата контроля>,
                  <код результата исполнения процедуры>,
                  <сообщение результата исполнения процедуры>);
```

Процедура ETL_MASS_ERROR_REGISTER

Процедура предназначена для массовой регистрации результата контроля качества данных в журнале контроля качества STG_LOG_ERRORS.

В рамках процедуры выполняются:

1. Контроль параметров:
Если P_SQL равен NULL, то процедура прерывает свою работу с кодом возврата -2 и сообщением «Не указан SQL-запрос для контроля качества данных».
2. Регистрируются записи в журнале контроля качества STG_LOG_ERRORS по результатам SQL-запроса.
3. Возвращаются:
 - a. Результат исполнения процедуры (P_RESULT)
 - b. Сообщение результата исполнения процедуры (P_RESULT_MSG)
 - c. Количество вставленных записей (P_INS_REC_COUNT)

Все программные исключения, которые могут возникнуть в ходе выполнения перехватываются и процедура возвращает данные исключения в параметрах исполнения результата.

Синтаксис

```
ETL_MASS_ERROR_REGISTER(p_sql, p_result, p_ins_rec_count)
```

Параметры***p_sql in TEXT***

Параметр принимает текст SQL-запроса, результаты которого записываются в журнал протокола контроля качества данных STG_LOG_ERRORS.

p_result in out number

Параметр возвращает значение кода результата исполнения данной процедуры.

Возможные значения:

- 0 – процедура отработала успешно
- (-1) – процедура отработала не успешно
- (-2) – не задан обязательный параметр запроса SQL

p_result_msg in out **TEXT**

Параметр возвращает сообщение результата исполнения данной процедуры.

p_ins_rec_count in out **number**

Параметр возвращает количество записей, вставленных в журнал протокола контроля качества данных STG_LOG_ERRORS по результатам запроса P_SQL.

Рекомендуемый пример вызова

```
ETL_mass_error_register(<SQL- запрос>,  
                        <код результата исполнения процедуры>,  
                        <сообщение результата исполнения процедуры>,  
                        <количество вставленных записей>);
```

3.5 Правила использования пользовательских этапов обработки и контроля данных

В рамках процесса переноса данных из Staging Area в Хранилище данных определены следующие этапы, в рамках которых могут исполняться прикладные процедуры обработки и контроля данных, как области Stage, так и области Хранилища:

1. GLOBAL_CHECK – выполнение общих пользовательских действий в области Stage **до** системной проверки (GLOBAL_SYCHK) в рамках сеанса импорта данных.
2. GLOBAL_BEFORE – выполнение общих пользовательских действий **до** загрузки данных в Хранилище в рамках сеанса импорта данных
3. DATABANK_BEFORE – выполнение пользовательских действий **до** загрузки данных в Хранилище в рамках сеанса импорта данных для определенного банка данных
4. DATABANK_AFTER – выполнение пользовательских действий **после** загрузки данных в Хранилище в рамках сеанса импорта данных для определенного банка данных
5. GLOBAL_AFTER – выполнение общих пользовательских действий **после** загрузки данных в Хранилище в рамках сеанса импорта данных

Такие процедуры должны быть зарегистрированы в [настройках правил обработки и контроля качества данных](#) для определенных этапа процесса и банка данных (для этапов DATABANK_BEFORE, DATABANK_AFTER).

3.5.1 Требования к прикладным процедурам, используемым в пользовательских этапах

Каждая прикладная процедура, используемая в рамках пользовательских этапов, должна иметь предопределенный набор входных/выходных параметров.

Обязательные параметры перечислены в следующей таблице:

№	Параметр	Тип данных	Обязательный	Тип параметра	Описание
1.	p_session	NUMBER(10)	Да	IN	Идентификатор сеанса загрузки.
2.	p_severity	NUMBER(10)	Нет	OUT	Максимальный уровень критичности
3.	p_protocol	CLOB	Нет	OUT	Протокол исполнения прикладной процедуры обработки и контроля качества

4. <прикладные параметры>

Значения predetermined параметров определяются в управляющей процедуре, в которой производится вызов прикладных процедур обработки, соответствующего этапа процесса загрузки данных из Staging Area в DWH.

Помимо predetermined параметров пользовательские процедуры обработки данных могут иметь собственные параметры прикладного назначения. Значения параметров прикладного назначения должны быть зарегистрированы в [журнала регистрации значений параметров](#).

3.5.2 Требования к запросу контроля качества данных для процедуры ETL_MASS_ERROR_REGISTER

Запрос контроля качества должен быть оформлен в текстовом виде и передан процедуру [ETL_MASS_ERROR_REGISTER](#) с помощью параметра P_SQL.

Данная процедура ожидает от запроса следующий состав полей:

1. ER_STG_BIGINT – BIGINT соответствующей записи таблицы ETL_S_*. Может быть указано значение NULL.
2. ER_DWH_BIGINT – BIGINT соответствующей записи соответствующей таблицы Хранилища данных. Может быть указано значение NULL.
3. ER_MSG - сообщение ошибки для конкретной записи соответствующей stage-таблицы или таблицы DWH. Может быть указано значение NULL.
4. ER_SEVERITY – уровень критичности ошибки для конкретной логики контроля

3.5.3 Пакетные переменные, разрешенные к использованию в прикладных процедурах

В рамках прикладных процедур возможно использование значений некоторых пакетных переменных для определения текущего состояния сеанса импорта данных.

При использовании **запрещается** изменение значений этих переменных внутри прикладных процедур.

Перечень таких пакетных переменных приведен в следующей таблице:

№	Переменная	Тип данных	Назначение переменной
1.	CURRENT_SESSION_ID	NUMBER(10)	<p>Переменная используется для определения ID сеанса в рамках процедур пакета импорта данных. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p> <p>Значение по умолчанию: NULL Устанавливается при регистрации сеанса импорта данных.</p>
2.	CURRENT_SESSION_CODE	TEXT(128)	<p>Переменная используется для определения кода сеанса в рамках процедур пакета импорта данных. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p> <p>Значение по умолчанию: NULL Устанавливается при регистрации сеанса импорта данных.</p>
3.	CURRENT_PHASE	TEXT(35)	<p>Переменная используется для определения текущей фазы процесса импорта данных в процедурах пакета импорта. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p>

			<p>Значение по умолчанию: NULL</p> <p>Устанавливается при регистрации этапа процесса импорта данных.</p>
4.	CURRENT_SESSION_PHASE_ID	NUMBER(10)	<p>Переменная используется для определения текущего этапа процесса импорта, зарегистрированного в журнале исполнения процесса импорта данных. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p> <p>Значение по умолчанию: NULL</p> <p>Устанавливается при регистрации этапа процесса импорта данных.</p>
5.	CURRENT_DATABANK	TEXT	<p>Переменная используется для определения текущего банка данных, с которым производится работа в рамках соответствующего этапа процесса импорта, зарегистрированного в журнале исполнения процесса импорта данных. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p> <p>Значение по умолчанию: NULL</p> <p>Устанавливается при регистрации этапа процесса импорта данных.</p>
6.	CURRENT_METAOBJ	TEXT	<p>Переменная используется для определения текущего метаобъекта банка данных, с которым производится работа в рамках соответствующего этапа процесса импорта, зарегистрированного в журнале исполнения процесса импорта данных.</p> <p>Значение по умолчанию: NULL</p> <p>Устанавливается при регистрации этапа процесса импорта данных.</p>
7.	CURRENT_LOADOBJ	NUMBER(10)	<p>Переменная используется для определения текущей регистрации метаобъекта, зарегистрированного в журнале импортируемых метаобъектов банков данных. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p> <p>Значение по умолчанию: NULL</p> <p>Устанавливается при регистрации этапа процесса импорта данных.</p>
8.	CURRENT_PROCNAME	TEXT	<p>Переменная используется для хранения SQL-имени текущей исполняемой процедуры. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p> <p>Значение по умолчанию: NULL</p> <p>Устанавливается при регистрации этапа процесса импорта данных.</p>
9.	CURRENT_CHECK_ID	NUMBER(10)	<p>Переменная используется для определения текущей исполняемой процедуры обработки и контроля качества данных. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».</p> <p>Значение по умолчанию: NULL</p>

			Устанавливается при регистрации прикладной процедуры и подтверждения исполнения прикладной процедуры
10.	CURRENT_CRITICAL_SEVERITY	NUMBER(1)	Переменные используются для определения уровней критичности выполняемой процедуры обработки и контроля качества данных. Также может применяться в прикладных процедурах обработки и контроля с указанием префикса «ETL_».
11.	CURRENT_ERROR_SEVERITY	NUMBER(1)	
12.	CURRENT_WARNING_SEVERITY	NUMBER(1)	
			Значение по умолчанию: NULL Устанавливается при регистрации прикладной процедуры и подтверждения исполнения прикладной процедуры
13.	CR_LF	TEXT	Константа предназначена для формирования возврата каретки и переноса строки в строковых выражениях. Значение по умолчанию: CHR(13) CHR(10)

3.5.4 Рекомендации к SQL-именам процедур

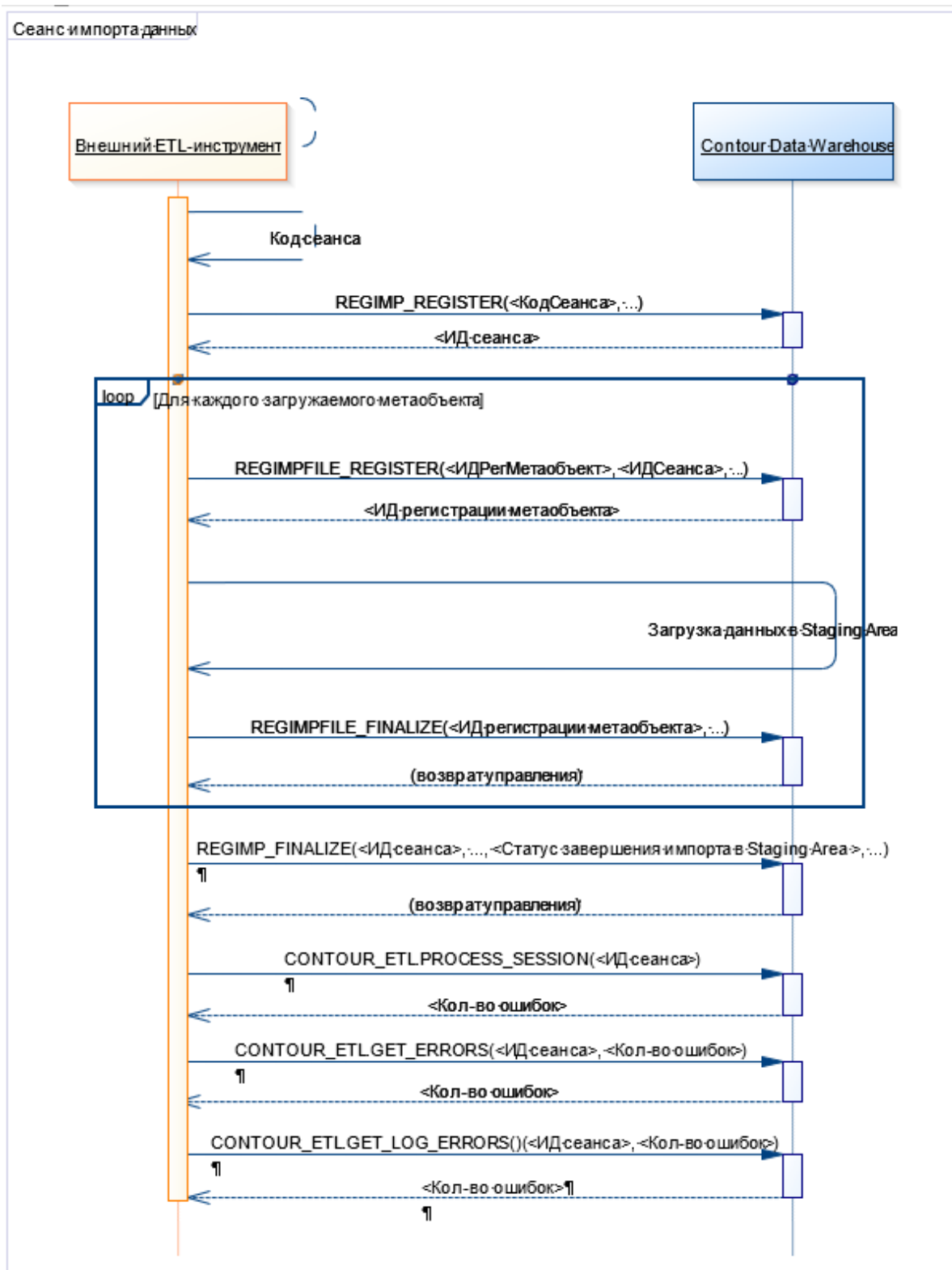
Рекомендации для префиксов SQL-имени прикладных процедур:

№	Префикс	Описание
1.	SGC_*	Для прикладных процедур вызов, которых предполагается производить на этапе GLOBAL_CHECK
2.	SGB_*	Для прикладных процедур вызов, которых предполагается производить на этапе GLOBAL_BEFORE
3.	SB_*	Для прикладных процедур вызов, которых предполагается производить на этапе DATABANK_BEFORE
4.	SP_*	Для прикладных процедур вызов, которых предполагается производить на этапе DATABANK_AFTER
5.	SGA_*	Для прикладных процедур вызов, которых предполагается производить на этапе GLOBAL_AFTER

Примечание: Если прикладные процедуры разрабатываются специалистами Заказчика, то рекомендуется указывать дополнительный префикс <код Заказчика>. Например таким образом: <код Заказчика>_SB#<целевое назначение>

3.6 Основной сценарий использования

3.6.1 Схема исполнения сеанса импорта данных



3.6.2 Описание процесса взаимодействия систем в рамках сеанса

Участники процесса:

1. Внешний ETL-инструмент
2. Система Contour Data Warehouse

№	Действия внешнего ETL-инструмента	№	Действия Contour Data Warehouse
1.	Старт сеанса импорта данных		
1.1	Формирует <Код сеанса>, объявляет <ИД сеанса>		
1.2	Вызывает процедуру регистрации сеанса REGIMP_REGISTER	1.2	Процедура регистрации сеанса:
		1.3	Получает и контролирует параметры
		1.4	Выполняет регистрацию сеанса в журнале сеансов. Формирует <ИД сеанса>
		1.5	Выполняет генерацию партиций таблиц Staging Area для сеанса
1.6	Получает <ИД сеанса>	1.6	Возвращает <ИД сеанса>
2.	Организует цикл импорта данных по метаобъектам. Для каждого метаобъекта выполняет:		
2.1	Объявляет <ИД рег. метаобъекта>		
2.2	Вызывает процедуру регистрации импорта метаобъекта REGIMPFIL_REGISTER для <ИД сеанса>	2.2	Процедура регистрации импорта метаобъекта:
		2.3	Получает и контролирует параметры
		2.4	Выполняет регистрацию импорта метаобъекта в рамках сеанса в журнале импорта данных по метаобъектам. Формирует <ИД рег. метаобъекта>
2.5	Получает <ИД рег. метаобъекта>	2.5	Возвращает <ИД рег. метаобъекта>
2.7	Выполняет импорт данных в таблицу метаобъекта в Staging Area		
2.7	Вызывает процедуру подтверждения импорта данных по метаобъекту REGIMPFIL_FINALIZE для <ИД рег. метаобъекта>	2.7	Процедура подтверждения завершения импорта данных по метаобъекту:
		2.8	Получает и контролирует параметры
		2.9	Обновляет запись для <ИД рег. метаобъекта>
2.10	Получает управление	2.10	Ничего не возвращает
2.11	Завершает цикл импорта данных по метаобъектам.		
3.	Вызывает процедуру подтверждения завершения сеанса REGIMP_FINALIZE для <ИД сеанса> со <статусом завершения сеанса>	3	Процедура подтверждения завершения сеанса импорта данных в Staging Area по сеансу:
		3.1	Получает и контролирует параметры
		3.2	Обновляется запись журнала сеансов в части атрибутов относящихся к загрузке в Staging Area.

3.3	Получает управление	3.3	Ничего не возвращает
4.	Вызывает процедуру управления переносом данных из Staging Area в Хранилище данных ETL_PROCESS_SESSION для <ИД сеанса>	4.	Ничего не возвращает
		4.1	Получает и контролирует параметры
		4.2	Исполняет пользовательские процедуры контроля и обработки данных до системных проверок
		4.3	Исполняет системные процедуры контроля данных
		4.4	Исполняет пользовательские процедуры контроля и обработки данных до переноса данных в ХД
		4.5	Переносит данные из Staging Area в ХД
		4.6	Исполняет пользовательские процедуры контроля и обработки данных после переноса данных в ХД
		4.7	Очищает область Staging Area для сеанса
4.8	Получает управление	4.8	Ничего не возвращает
5.	Вызывает процедуру ETL_GET_ERRORS для определения <количества ошибок данных> для <ИД сеанса>	5.	Процедура определения количества ошибок данных:
		5.1	Получает <ИД сеанса>
5.2	Получает <количество ошибок данных>	5.2	Возвращает <количество ошибок данных>
6.	Вызывает процедуру ETL_GET_LOG_ERRORS для определения <количества ошибок процесса> для <ИД сеанса>	6.	Процедура определения количества ошибок процесса:
		6.1	Получает <ИД сеанса>
6.2	Получает <количество ошибок процесса>	6.2	Возвращает <количество ошибок процесса>
7.	Определяет <статус завершения сеанса>. Если <количества ошибок процесса> + <количество ошибок данных> > 0, то <статус завершения сеанса> = 'N', иначе <статус завершения сеанса> = 'Y',		
8.	Конец взаимодействия по сеансу		

Приложение 1. Пример xml-описания метаобъекта

```
<?Contour version="5.0" ?>
<Сеанс>
  <Приложение>
    <ПапкиКаталогаБанковДанных Количество="1">
      <ПапкаКаталогаБанковДанных ОбъектКаталога="С">
        <Название>ФХД для банков РФ</Название>
        <ПапкиКаталогаБанковДанных Количество="1">
          <ПапкаКаталогаБанковДанных ОбъектКаталога="С">
            <Название>Бухгалтерский учет КО</Название>
            <ПапкиКаталогаБанковДанных Количество="1">
              <ПапкаКаталогаБанковДанных ОбъектКаталога="С">
                <Название>Аналитический учет</Название>
                <ПапкиКаталогаБанковДанных Количество="1">
                  <ПапкаКаталогаБанковДанных ОбъектКаталога="С">
                    <Название>Кредиты юридических лиц</Название>
                    <БанкиДанных Количество="1">
                      <БанкДанных Код="КредитыЮЛ">
                        <СистемноеПриложение>UserBanks</СистемноеПриложение>
                        <Название>Кредиты ЮЛ</Название>
                        <SQLИмя>DL_LOANJ</SQLИмя>
                        <Историзация>Y</Историзация>
                        <ОбластьВременногоХранения>Y</ОбластьВременногоХранения>
                        <ТипИдентификатора>I</ТипИдентификатора>
                        <ДлинаКода>50</ДлинаКода>
                        <Защита>N</Защита>
                      <ГруппыОбъектов Количество="1">
                        <ГруппаОбъектов Код="КредитыЮЛФ303">
                          <СистемноеПриложение>UserBanks</СистемноеПриложение>
                          <Название>Классификация кредитов ЮЛ Ф303</Название>
                          <ИмяТаблицы>DL_LOANJ_G_F303</ИмяТаблицы>
                          <Историзация>Y</Историзация>
                          <ОбластьВременногоХранения>N</ОбластьВременногоХранения>
                          <Защита>N</Защита>
                        <АтрибутыМетаобъекта Количество="4">
                          <Заменить>0</Заменить>
                          <АтрибутМетаобъекта Код="LOAN_KIND">
                            <Название>Вид ссуды ф.303</Название>
```



```

<ИмяПоля>LOAN_KIND</ИмяПоля>
<Обязательность>Y</Обязательность>
<РазметкаАтрибутов>U</РазметкаАтрибутов>
<КонтрольЦелостности>N</КонтрольЦелостности>
<СсылкаНаБанкДанных>ENM_303_LOAN_KIND</СсылкаНаБанкДанных>
<КодДомена>udf_da_code</КодДомена>
<ДлинаТипаДанных>10</ДлинаТипаДанных>
<Защита>N</Защита>
</АтрибутМетаобъекта>
<АтрибутМетаобъекта Код="КодКлассиф">
  <Название>Код классификации</Название>
  <ИмяПоля>CLS_CODE</ИмяПоля>
  <Обязательность>N</Обязательность>
  <РазметкаАтрибутов>U</РазметкаАтрибутов>
  <КонтрольЦелостности>N</КонтрольЦелостности>
  <КодДомена>udf_da_objcode</КодДомена>
  <Защита>N</Защита>
</АтрибутМетаобъекта>
<АтрибутМетаобъекта Код="ПравилоКлассиф">
  <Название>Правило классификации</Название>
  <ИмяПоля>CLS_RULE_ID</ИмяПоля>
  <Обязательность>N</Обязательность>
  <РазметкаАтрибутов>U</РазметкаАтрибутов>
  <КонтрольЦелостности>N</КонтрольЦелостности>
  <КодДомена>id_int</КодДомена>
  <Защита>N</Защита>
</АтрибутМетаобъекта>
<АтрибутМетаобъекта Код="ПризнакКлассиф">
  <Название>Признак классификации</Название>
  <ИмяПоля>CLS_PROCESSING_KIND</ИмяПоля>
  <Обязательность>Y</Обязательность>
  <РазметкаАтрибутов>U</РазметкаАтрибутов>
  <КонтрольЦелостности>N</КонтрольЦелостности>
  <СсылкаНаБанкДанных>СпрВидОбработки</СсылкаНаБанкДанных>
  <КодДомена>udf_da_code</КодДомена>
  <ДлинаТипаДанных>1</ДлинаТипаДанных>
  <Защита>N</Защита>
</АтрибутМетаобъекта>
</АтрибутыМетаобъекта>
</ГруппаОбъектов>
</ГруппыОбъектов>

```

```

<АтрибутыМетаобъекта Количество="1">
  <Заменить>0</Заменить>
  <АтрибутМетаобъекта Код="НомерДоговора">
    <Название>Номер договора</Название>
    <Примечание>Номер кредитного договора.</Примечание>
    <ИмяПоля>CONTRACT_NO</ИмяПоля>
    <Обязательность>Y</Обязательность>
    <РазметкаАтрибутов>U</РазметкаАтрибутов>
    <КонтрольЦелостности>N</КонтрольЦелостности>
    <КодДомена>udf_da_code</КодДомена>
    <Защита>N</Защита>
  </АтрибутМетаобъекта>
</АтрибутыМетаобъекта>
</БанкДанных>
</БанкиДанных>
</ПапкаКаталогаБанковДанных>
</ПапкиКаталогаБанковДанных>
</ПапкаКаталогаБанковДанных>
</ПапкиКаталогаБанковДанных>
</ПапкаКаталогаБанковДанных>
</ПапкиКаталогаБанковДанных>
</ПапкаКаталогаБанковДанных>
</ПапкиКаталогаБанковДанных>
</Приложение>
</Сеанс>

```

Приложение 2. XSD-схема словаря метаобъектов ХД

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="https://www.InterSoftLab.ru/xml/isbints-1.3"
xmlns="https://www.InterSoftLab.ru/xml/isbints-1.3">
  <xs:element type="СеансType" name="Сеанс" />
  <xs:complexType name="СеансType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element type="ПриложениеType" name="Приложение" />
    </xs:choice>
  </xs:complexType>
  <xs:element type="ПриложениеType" name="Приложение" />
  <xs:complexType name="ПриложениеType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="СерверБД">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="100" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="БазаДанных">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="100" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element type="ТипыДанныхType" name="ТипыДанных" />
      <xs:element type="ПапкиКаталогаБанковДанныхType" name="ПапкиКаталогаБанковДанных" />
    </xs:choice>
  </xs:complexType>
  <xs:element type="ТипыДанныхType" name="ТипыДанных" />
  <xs:complexType name="ТипыДанныхType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element type="ТипДанныхType" name="ТипДанных" />
    </xs:choice>
    <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
  </xs:complexType>
```

```

<xs:element name="ПапкиКаталогаБанковДанных" type="ПапкиКаталогаБанковДанныхType" />
<xs:complexType name="ПапкиКаталогаБанковДанныхType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="ПапкаКаталогаБанковДанных" type="ПапкаКаталогаБанковДанныхType" />
  </xs:choice>
  <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="ТипДанныхType" name="ТипДанных" />
<xs:complexType name="ТипДанныхType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element type="xs:integer" name="ТипSQL" />
    <xs:element type="xs:integer" name="Длина" />
    <xs:element type="xs:integer" name="Точность" />
    <xs:element name="ПерменнаяДлина">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="1" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Статус">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="1" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="НазваниеТипаSQL">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Код">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
</xs:complexType>

```

```

        <xs:element name="Название">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="80" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Название">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="80" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:element name="ПапкаКаталогаБанковДанных" type="ПапкаКаталогаБанковДанныхType" />
<xs:complexType name="ПапкаКаталогаБанковДанныхType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Название">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="120" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="ПапкиКаталогаБанковДанных" type="ПапкиКаталогаБанковДанныхType" />
        <xs:element type="БанкиДанныхType" name="БанкиДанных" />
        <xs:element name="ОбъектКаталога">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="1" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
</xs:complexType>

```

```

        </xs:element>
    </xs:choice>
    <xs:attribute name="ОбъектКаталога">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:element type="БанкиДанныхType" name="БанкиДанных" />
<xs:complexType name="БанкиДанныхType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element type="АналитическаяВыборкаType"
name="АналитическаяВыборка" />
        <xs:element type="ВидКлассификацииType" name="ВидКлассификации"
/>
        <xs:element type="БанкДанныхType" name="БанкДанных" />
    </xs:choice>
    <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="АналитическаяВыборкаType" name="АналитическаяВыборка" />
<xs:complexType name="АналитическаяВыборкаType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="СистемноеПриложение">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Название">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="120" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="SQLИмя">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
</xs:complexType>

```

```

        </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="ДлинаКода" />
    <xs:element
        type="АтрибутыМетаобъектаType"
name="АтрибутыМетаобъекта" />
    <xs:element name="ТекстЗапроса">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="7000" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element
        type="АтрибутыАналитическойВыборкиType"
name="АтрибутыАналитическойВыборки" />
    <xs:element
        type="ПараметрыВыборкиType"
name="ПараметрыВыборки" />
    <xs:element name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:choice>
<xs:attribute name="Код">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:element type="ВидКлассификацииType" name="ВидКлассификации" />
<xs:complexType name="ВидКлассификацииType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="СистемноеПриложение">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Название">
            <xs:simpleType>

```

```

        <xs:restriction base="xs:string">
            <xs:maxLength value="120" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="SQLИмя">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element type="xs:integer" name="ДлинаКода" />
<xs:element
    name="АтрибутыМетаобъекта" />
    type="АтрибутыМетаобъектаType"
<xs:element name="ТекстЗапроса">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="7000" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element
    name="АтрибутыАналитическойВыборки" />
    type="АтрибутыАналитическойВыборкиType"
<xs:element
    name="ПараметрыВыборки" />
    type="ПараметрыВыборкиType"
<xs:element name="Группа">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="ОбъектКлассификации">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element
    name="АтрибутыГрупповойТаблицы" />
    type="АтрибутыГрупповойТаблицыType"
<xs:element
    name="АтрибутыБазовойТаблицаСправочникаПравил" />
    type="АтрибутыБазовойТаблицаСправочникаПравилType"

```



```

        <xs:element name="Код">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
    <xs:attribute name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
<xs:element type="БанкДанныхType" name="БанкДанных" />
<xs:complexType name="БанкДанныхType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="СистемноеПриложение">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Название">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="120" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Примечание">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="1024" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="SQLИмя">
            <xs:simpleType>
                <xs:restriction base="xs:string">

```

```

        <xs:maxLength value="35" />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Структура">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="20" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Историзация">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="20" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="ОбластьВременногоХранения">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="ТипИдентификатора">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="10" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element type="xs:integer" name="ДлинаКода" />
<xs:element name="Удалить">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Защита">
    <xs:simpleType>
        <xs:restriction base="xs:string">

```

```

                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element type="ТипыОбъектовType" name="ТипыОбъектов" />
    <xs:element type="ГруппыОбъектовType" name="ГруппыОбъектов" />
    <xs:element
        type="АтрибутыМетаобъектаType"
name="АтрибутыМетаобъекта" />
    <xs:element type="ИерархииОбъектовType" name="ИерархииОбъектов"
/>

    <xs:element type="НавигаторыType" name="Навигаторы" />
    <xs:element name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:choice>
<xs:attribute name="Код">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:element type="АтрибутыМетаобъектаType" name="АтрибутыМетаобъекта" />
<xs:complexType name="АтрибутыМетаобъектаType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Метаобъект">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
</xs:complexType>

```

```

        </xs:element>
        <xs:element type="xs:integer" name="Заменить" />
        <xs:element type="АтрибутАналитическойВыборкиType"
name="АтрибутАналитическойВыборки" />
        <xs:element type="АтрибутМетаобъектаType"
name="АтрибутМетаобъекта" />
        <xs:element type="ПараметрВыборкиType" name="ПараметрВыборки"
/>
    </xs:choice>
    <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="АтрибутыАналитическойВыборкиType"
name="АтрибутыАналитическойВыборки" />
<xs:complexType name="АтрибутыАналитическойВыборкиType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Метаобъект">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element type="xs:integer" name="Заменить" />
        <xs:element type="АтрибутАналитическойВыборкиType"
name="АтрибутАналитическойВыборки" />
    </xs:choice>
    <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="ПараметрыВыборкиType" name="ПараметрыВыборки" />
<xs:complexType name="ПараметрыВыборкиType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>

```

```

        </xs:element>
        <xs:element name="Метаобъект">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="РазметкаАтрибутов">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="512" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element type="xs:integer" name="Заменить" />
        <xs:element type="ПараметрВыборкиType" name="ПараметрВыборки"
/>

        </xs:choice>
        <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
    </xs:complexType>
    <xs:element name="АтрибутыГрупповойТаблицы" type="АтрибутыГрупповойТаблицыType" />
    <xs:complexType name="АтрибутыГрупповойТаблицыType">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="Категория">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:maxLength value="35" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Метаобъект">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:maxLength value="35" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element type="xs:integer" name="Заменить" />
            <xs:element type="АтрибутАналитическойВыборкиType"
name="АтрибутАналитическойВыборки" />
        </xs:choice>
        <xs:attribute type="xs:nonNegativeInteger" name="Количество" />

```

```

</xs:complexType>
<xs:element type="АтрибутыБазовойТаблицаСправочникаПравилType"
name="АтрибутыБазовойТаблицаСправочникаПравил" />
<xs:complexType name="АтрибутыБазовойТаблицаСправочникаПравилType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Категория">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Метаобъект">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="Заменить" />
    <xs:element type="АтрибутАналитическойВыборкиType"
name="АтрибутАналитическойВыборки" />
    <xs:element type="АтрибутМетаобъектаType"
name="АтрибутМетаобъекта" />
    <xs:element type="ПараметрВыборкиType" name="ПараметрВыборки"
/>
  </xs:choice>
  <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="ТипыОбъектовType" name="ТипыОбъектов" />
<xs:complexType name="ТипыОбъектовType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Категория">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Вид">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="1" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
</xs:complexType>

```

```

        </xs:simpleType>
    </xs:element>
    <xs:element type="ТипОбъектовType" name="ТипОбъектов" />
</xs:choice>
    <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="ГруппыОбъектовType" name="ГруппыОбъектов" />
<xs:complexType name="ГруппыОбъектовType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element type="ГруппаОбъектовType" name="ГруппаОбъектов" />
    </xs:choice>
    <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="ИерархииОбъектовType" name="ИерархииОбъектов" />
<xs:complexType name="ИерархииОбъектовType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element type="ИерархияОбъектовType" name="ИерархияОбъектов"
/>

    </xs:choice>
    <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="НавигаторыType" name="Навигаторы" />
<xs:complexType name="НавигаторыType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>

```

```

        </xs:simpleType>
    </xs:element>
    <xs:element type="НавигаторType" name="Навигатор" />
</xs:choice>
<xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element name="АтрибутАналитическойВыборки" type="АтрибутАналитическойВыборкиType" />
<xs:complexType name="АтрибутАналитическойВыборкиType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Название">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="120" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Примечание">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="1024" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="ИмяПоля">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="30" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Метаобъект">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
</xs:complexType>

```



```

        </xs:simpleType>
    </xs:element>
    <xs:element name="Обязательность">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="РазметкаАтрибутов">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="512" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="КонтрольЦелостности">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="СсылкаНаБанкДанных">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="КодДомена">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="ДлинаТипаДанных" />
    <xs:element name="ЗначениеПоУмолчанию">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1024" />
            </xs:restriction>

```

```

        </xs:simpleType>
    </xs:element>
    <xs:element name="Защита">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:choice>
<xs:attribute name="Код">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:element type="АтрибутМетаобъектаType" name="АтрибутМетаобъекта" />
<xs:complexType name="АтрибутМетаобъектаType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Название">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="120" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Примечание">

```

```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1024" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ИмяПоля">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="30" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Метаобъект">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Обязательность">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="РазметкаАтрибутов">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="512" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="КонтрольЦелостности">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="СсылкаНаБанкДанных">
        <xs:simpleType>

```

```

        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="КодДомена">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element type="xs:integer" name="ДлинаТипаДанных" />
<xs:element name="ЗначениеПоУмолчанию">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1024" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Защита">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Код">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:choice>
<xs:attribute name="Код">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>

```

```

<xs:element type="ПараметрВыборкиType" name="ПараметрВыборки" />
<xs:complexType name="ПараметрВыборкиType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Код">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Название">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="255" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="АльтернативноеИмя">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="255" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="Позиция" />
    <xs:element name="ФормулаТолькоЧтение">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="8000" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="ФормулаУмолчение">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="8000" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Формула">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="8000" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
</xs:complexType>

```

```

        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Видимость">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Алгоритм">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Категория">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Метаобъект">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="КодТипа">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Защита">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>

```

```

        </xs:simpleType>
      </xs:element>
      <xs:element name="РазметкаАтрибутов">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="512" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Обязательность">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element
name="ОпцииПараметраВыборки" />
type="ОпцииПараметраВыборкиType"
    </xs:choice>
  </xs:complexType>
  <xs:element type="ТипОбъектовType" name="ТипОбъектов" />
  <xs:complexType name="ТипОбъектовType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="Категория">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="СистемноеПриложение">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Название">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="120" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:choice>
  </xs:complexType>

```

```

</xs:element>
<xs:element name="Примечание">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="1024" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="ИмяТаблицы">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="20" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Вид">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="1" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Историзация">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="1" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="ОбластьВременногоХранения">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="1" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Защита">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="1" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```



```

                                type="АтрибутыМетаобъектаType"
name="АтрибутыМетаобъекта" />
    <xs:element name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:choice>
<xs:attribute name="Код">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="35" />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:element type="ГруппаОбъектовType" name="ГруппаОбъектов" />
<xs:complexType name="ГруппаОбъектовType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="СистемноеПриложение">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Название">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="120" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Примечание">

```

```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1024" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ИмяТаблицы">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="20" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Историзация">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ОбластьВременногоХранения">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Защита">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="АтрибутыМетаобъекта" type="АтрибутыМетаобъектаType" />
    <xs:element name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>

```

```

</xs:choice>
<xs:attribute name="Код">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="35" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:element type="ИерархияОбъектовType" name="ИерархияОбъектов" />
<xs:complexType name="ИерархияОбъектовType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Категория">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Код">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Название">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="120" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="SQLИмя">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="3" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Статус">
      <xs:simpleType>
        <xs:restriction base="xs:string">

```

```

        <xs:maxLength value="10" />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="ОбластьВременногоХранения">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="ЗащитаМетаобъекта">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="ОчищатьПриЗагрузке">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element type="УровниИерархииType" name="УровниИерархии" />
</xs:choice>
</xs:complexType>
<xs:element type="НавигаторType" name="Навигатор" />
<xs:complexType name="НавигаторType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="СистемноеПриложение">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:choice>
</xs:complexType>

```

```

        </xs:simpleType>
    </xs:element>
    <xs:element name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Название">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="120" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="SQLИмяИндекса">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="3" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Группа">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Тип">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="Компрессия" />
    <xs:element name="Статус">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="10" />
            </xs:restriction>

```

```

        </xs:simpleType>
    </xs:element>
    <xs:element name="Защита">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Уникальность">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="1" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element
        name="АтрибутыНавигатора" />
        type="АтрибутыНавигатораType"
    </xs:choice>
</xs:complexType>
<xs:element type="ОпцииПараметраВыборкиType" name="ОпцииПараметраВыборки" />
<xs:complexType name="ОпцииПараметраВыборкиType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Категория">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Параметр">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="35" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element
            name="ОпцияПараметраВыборки" />
            type="ОпцияПараметраВыборкиType"
        </xs:choice>
        <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
    </xs:complexType>
<xs:element type="УровниИерархииType" name="УровниИерархии" />

```

```

<xs:complexType name="УровниИерархииType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Категория">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Иерархия">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="Заменить" />
    <xs:element type="УровеньИерархииType" name="УровеньИерархии" />
  </xs:choice>
  <xs:attribute type="xs:nonNegativeInteger" name="Количество" />
</xs:complexType>
<xs:element type="АтрибутыНавигатораType" name="АтрибутыНавигатора" />
<xs:complexType name="АтрибутыНавигатораType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Категория">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Навигатор">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="Заменить" />
    <xs:element type="АтрибутНавигатораType" name="АтрибутНавигатора"
  />
  </xs:choice>
  <xs:attribute type="xs:nonNegativeInteger" name="Количество" />

```

```

</xs:complexType >
<xs:element type="ОпцияПараметраВыборкиType" name="ОпцияПараметраВыборки" />
<xs:complexType name="ОпцияПараметраВыборкиType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Категория">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Код">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Параметр">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Описание">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="255" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Значение">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="255" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
</xs:complexType >
<xs:element type="УровеньИерархииType" name="УровеньИерархии" />
<xs:complexType name="УровеньИерархииType">

```



```

<xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element name="Категория">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="35" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element type="xs:integer" name="Код" />
  <xs:element name="Название">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="120" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Иерархия">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="35" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Защита">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="1" />
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:choice>
</xs:complexType>
<xs:element type="АтрибутНавигатораType" name="АтрибутНавигатора" />
<xs:complexType name="АтрибутНавигатораType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Категория">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="35" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:choice>
  <xs:element name="Навигатор">

```

```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Код">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="НавигаторСсылка">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element type="xs:integer" name="Позиция" />
    <xs:element name="Метаобъект">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
</xs:choice>
</xs:complexType>
</xs:schema>

```

Приложение 3. Пример сценария создания справочников

```
isdbinst version="1.0"
```

```
xml "СпрТипПериода.xml"
```

```
data "СпрТипПериода.data.kdl"
```

```
xml "ВидСчетДенЕдин.xml"
```

```
data "ВидСчетДенЕдин.data.kdl"
```

```
xml "ОКВ_ОКЕИ.xml"
```

```
data "ОКВ_ОКЕИ.data.kdl"

xml "РегионМира.xml"
data "РегионМира.data.kdl"
xml "Страны.xml"
data "Страны.data.kdl"
xml "СоставЕвразоны.xml"
data "СоставЕвразоны.data.kdl"

xml "ВалНедСтран.xml"
data "ВалНедСтран.data.kdl"
xml "КурсыВалют.xml"
data "КурсыВалют.data.kdl"
xml "КотировкиВалют.xml"
data "КотировкиВалют.data.kdl"

sql "so_get_calendar.sql"
xml "НерабочиеДни.xml"
data "НерабочиеДни.data.kdl"

xml "СпрСтатусСчет.xml"
data "СпрСтатусСчет.data.kdl"
xml "СпрСторонаСчет.xml"
data "СпрСторонаСчет.data.kdl"
```

Приложение 4. Пример сложного сценария импорта метаданных

```

isdbinst version="1.0"

sql "lgjorngroup.sql"
sql "lgjournal.sql"
sql "lgmessage.sql"

sql "lgctx_set_current_journal.sql"
sql "lgctx_get_current_journal.sql"
sql "lgctx_set_current_message.sql"
sql "lgctx_get_current_message.sql"

sql "lgjournal_close_internal.sql"
sql "lgjournal_create_internal.sql"
sql "lgmessage_write_internal.sql"
sql "lgmessage_update_internal.sql"

if {
    bool-query "select capp_is_pgpro_enterprise()"
    sql "lgjournal_close_pro.sql"
    sql "lgjournal_create_pro.sql"
    sql "lgmessage_write_pro.sql"
    sql "lgmessage_update_pro.sql"
}
elif{
    bool-query "select not exists(select from pg_roles where rolname='contour_logger' and
rolcanlogin)"
    abort "Не создан служебный пользователь contour_logger. Установка
прекращена."
}
else{
    sql "lgjournal_close_proxy.sql"
    sql "lgjournal_create_proxy.sql"
    sql "lgmessage_write_proxy.sql"
    sql "lgmessage_update_proxy.sql"
}

sql "lgjournal_close.sql"
sql "lgjournal_create.sql"
sql "lgjournal_open.sql"
sql "lgmessage_write.sql"

```