



Язык FL OSQL

DML DDL ADM

В документе содержится полное описание внутреннего языка запросов OSQL, являющегося объектным расширением языка SQL

Версия 2010

ВВЕДЕНИЕ	3
1. СИНТАКСИЧЕСКИЕ ОБОЗНАЧЕНИЯ	3
2. ДОМЕНЫ	5
3. КОМАНДЫ ODDL	5
3.1. CREATE TYPE (AA)	5
3.2. CREATE VIRTUAL TYPE (ART).....	8
3.3. ALTER TYPE (AA)	9
3.4. ALTER TYPE SET STORE_POLICY	11
3.5. DROP TYPE (AA)	12
3.6. DROP VIRTUAL TYPE (ART).....	12
3.7. DESCRIBE TYPE	13
4. КОМАНДЫ ODML	13
4.1. CREATE OBJECT (AA).....	13
4.2. UPDATE OBJECT (AA)	14
4.3. DELETE OBJECT (AA)	15
4.4. SELECT (AA)	15
4.4.1. ПРИМЕРЫ ЗАПРОСОВ ПО ПОЛЬЗОВАТЕЛЯМ И ГРУППАМ	19
4.5. GET AUDIT (AA)	20
5. КОМАНДЫ ADM	21
5.1. CREATE USER.....	21
5.2. ALTER USER.....	22
5.3. DELETE USER ().....	23
5.4. CREATE GROUP (ART).....	23
5.5. ALTER GROUP (ART).....	24
5.6. DELETE GROUP (ART).....	26
5.7. GET USERS FROM GROUP (ART)	26
5.8. ADD USER TO GROUP ().....	27
5.9. REMOVE USER FROM GROUP ()	28
5.10. ADD GROUP TO GROUP (ART)	29
5.11. REMOVE GROUP FROM GROUP (ART)	29
5.12. RECALCULATE CACHE ()	30
5.13. PURGE AUDIT ().....	31
5.14. PURGE STATISTICS ()	32
5.15. EXECUTE SQL ().....	33
6. КОМАНДЫ PREPARED STATEMENT	33

ВВЕДЕНИЕ

Целевой аудиторией этого документа являются администраторы и разработчики, которые должны иметь базовые понятия о программировании на java, инсталляции и установке веб-приложений, веб-программировании, трехзвенной архитектуре.

1. СИНТАКСИЧЕСКИЕ ОБОЗНАЧЕНИЯ

В следующей таблице перечислены и описаны соглашения, которые используются в диаграммах синтаксиса в справочнике по серверным командам.

Таблица 1

Соглашение	Используется для
ВЕРХНИЙ РЕГИСТР	Ключевые слова
<i>курсив</i>	Пользовательские параметры синтаксиса.
полужирный	Имена баз данных, таблиц, столбцов, индексов, хранимых процедур, программ, типов данных и текст должны вводиться в точном соответствии с примером.
подчеркнутый	Указывает значение по умолчанию, которое применяется, когда в инструкции пропущено предложение, содержащее подчеркнутое значение.
(вертикальная черта)	Разделяет элементы синтаксиса внутри квадратных или фигурных скобок. Может быть использован только один из этих элементов.
[] (квадратные скобки)	Необязательные элементы синтаксиса. Скобки вводить не следует.
{ } (фигурные скобки)	Обязательные элементы синтаксиса. Скобки вводить не следует.
[, ... n]	Указывает на то, что предшествующий элемент можно повторить n раз. Отдельные вхождения элемента разделяются запятыми.
[... n]	Указывает на то, что предшествующий элемент можно повторить n раз. Отдельные вхождения элемента разделяются пробелами.
[;]	Необязательный признак конца инструкции.

	Скобки вводить не следует.
<метка> ::=	Имя синтаксического блока. Данное соглашение используется для группирования и маркировки длинных частей синтаксиса или синтаксического блока, который может использоваться в нескольких местах в рамках одной инструкции. Каждое место, в котором может быть использован синтаксический блок, обозначается меткой, заключенной в двойные угловые скобки: <метка>.

Словарь данных - Под словарем данных репозитория понимается информация о метаданных, т.е. об описании используемых системой типов и объектов.

2. ДОМЕНЫ

INTEGER (ЦЕЛОЕ) – число, максимум 10 знаков;

ID (ИД) – идентификатор;

STRING (размер) (СТРОКА) – строка заданного размера;

NUMBER - количество знаков до запятой, количество знаков после запятой (ДРОБНОЕ) – нецелое число;

DATE (ДАТА) – дата и время (дд.ММ.гггг чч.мм.сс.сссс);

BOOLEAN – логический тип;

3. КОМАНДЫ ODDL

Под командами ODDL понимаются команды, служащие для формирования словаря данных (object data definition language). Команды для работы со словарём данных (Data Dictionary) служат для создания/модификации/удаления типов объектов, изменения/добавления/удаления атрибутов, установки иных параметров определенных типов словаря данных.

Важным ограничением является то, что системные типы словаря данных нельзя модифицировать или удалять: это разрушает целостность системы.

3.1. CREATE TYPE (AA)

CREATE TYPE – Создает новый объектный тип, определяемый пользователем, в репозитории. В дополнение к созданию информации о создаваемом типе в словаре данных создаются и все необходимые таблицы и представления (view), требуемые для функционирования экземпляров объектного типа в репозитории.

Синтаксис

```
CREATE TYPE type_name
{
    SUPERTYPE supertype_name | NONE
    [ SECURITY NONE | ACL [ TEMPLATE <идентификатор> ] | FIELDS ]
    [<блок атрибутов>]
    [<блок функциональности>]
    [<блок ограничений>]
};
<блок атрибутов> ::= имя атрибута <домен>
<домен> ::= INTEGER | ID | STRING | NUMBER | DATE | BOOLEAN
<блок функциональности> ::= FUNCTIONALITY <список параметров> [ , ... n ]
<список параметров> ::=
    HIERARCHY { NONE | ENABLED | CONTAINER }
    VERSIONING { ENABLED | NONE }
    CHECKOUTABLE { ENABLED | NONE }
    CONTENT { ENABLED | NONE }
<блок ограничений> ::=
    UPDATEOBJECT { DIRECT | SYSTEM_ONLY | API_ONLY }
    CREATEOBJECT { DIRECT | SYSTEM_ONLY | API_ONLY }
```

```
ALERTYPE { DIRECT | SYSTEM_ONLY }
DROPTYPE { DIRECT | SYSTEM_ONLY }
DELETEOBJECT { DIRECT | SYSTEM_ONLY | API_ONLY }
```

Аргументы

type_name

Имя псевдонима типа данных или определяемого пользователем типа данных.

supertype

Указывается имя родительского типа при наследовании, без наследования указывается NONE.

security

Указывается налагаемая на объект безопасность. NONE – если безопасность отсутствует. ACL – безопасность налагается только на объекты. FIELDS – безопасность налагается на объекты и на группы атрибутов в объекте. Параметр необязательный. При наследовании – тип безопасности потомка совпадает с типом безопасности родителя. При отсутствии наследования и указания типа безопасности – назначается отсутствие безопасности.

блок атрибутов

Перечисление атрибутов типа. Определение атрибута состоит из указания имени атрибута и указания домена атрибута. Атрибуты могут быть репитинговые (не могут быть у security fields)

Доступные домены:

- INTEGER (ЦЕЛОЕ) – число, максимум 10 знаков;
- ID (ИД) – идентификатор;
- STRING (размер) (СТРОКА) – строка заданного размера;
- NUMBER - количество знаков до запятой, количество знаков после запятой (ДРОБНОЕ) – нецелое число;
- DATE (ДАТА) – дата и время (дд.ММ.гггг чч.мм.сс.сссс);
- BOOLEAN – логический тип;

блок функциональности

определяет дополнительную функциональность создаваемого типа.

Параметры:

HIERARCHY { NONE | ENABLED | CONTAINER }

ИЕРАРХИЯ { НЕТ | ВКЛЮЧЕНА | КОНТЕЙНЕР };

VERSIONING { ENABLED | NONE }

возможность работы с версией объекта;

CHECKOUTABLE { ENABLED | NONE }

возможность блокирования объекта. К нему можно применять команды CHECKOUT, CHECKIN, UNDOCHECKOUT;

CONTENT {ENABLED | NONE }

возможность приложить файл к объекту типа;

блок ограничений

Список ограничений, накладываемый на тип.

Параметры:

UPDATEOBJECT { DIRECT | SYSTEM_ONLY | API_ONLY }

ограничение на изменение объектов типа;

CREATEOBJECT { DIRECT | SYSTEM_ONLY | API_ONLY }

ограничение на создание объектов типа;

ALERTTYPE { DIRECT | SYSTEM_ONLY }

ограничения на изменение типа;

DROPTYPE { DIRECT | SYSTEM_ONLY }

ограничения на удаление типа;

DELETEOBJECT { DIRECT | SYSTEM_ONLY | API_ONLY }

ограничения на удаление объектов типа.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Создаёт таблицы, заносит информацию о типе в SYS_DD_TYPE и информацию об атрибутах в SYS_DD_ATTRS

Ожидаемый результат

True/exception

Возможные ошибки

Атрибуты с одинаковым названием (в том числе наследуемые).

Пример:

```
CREATE TYPE type_a1 SUPERTYPE NONE SECURITY NONE (A1 INTEGER, B2 STRING(12))  
FUNCTIONALITY CONTENT ENABLED
```

DBCommand

org.corolla.server.datadictionary.management.commands.oddl.CreateType

3.2. CREATE VIRTUAL TYPE (ART)

CREATE VIRTUAL TYPE – Создает виртуальный тип, ссылающийся на таблицу (или представление-view), представляющую данные одной или более таблиц, определяемых в выражении SELECT.

Создание виртуального типа представляется целесообразным в том случае, если имеется необходимость использовать внешние по отношению к системе наборы данных (таблицы), такие как: справочники, ведущиеся во внешней системе, представления (view), возвращающие предопределенный набор данных.

Работа с виртуальным типом ограничена только выражением SELECT, иные операции манипулирования данными над виртуальным типом не предусмотрены.

Синтаксис

```
CREATE VIRTUAL TYPE type_name AS select_statement [ ; ]
```

select_statement

Инструкция SELECT, определяющая представление. В этой инструкции можно указывать более одной таблицы и другие представления. Для выбора объектов, указанных в предложении SELECT создаваемого представления, необходимы соответствующие разрешения.

Представление не обязательно является простым подмножеством строк и столбцов одной конкретной таблицы. С помощью предложения SELECT можно создавать представление, использующее более одной таблицы, или другие представления любой степени сложности.

При использовании в определении индексированного представления инструкция SELECT должна содержать указание одной таблицы или соединять инструкцией JOIN несколько таблиц с необязательной статистической обработкой.

type_name

Имя псевдонима типа данных или определяемого пользователем типа данных;

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Создаёт таблицы, заносит информацию о типе в SYS_DD_TYPE *Ожидаемый результат*

Ожидаемый результат

True/exception

Возможные ошибки

Атрибуты с одинаковым названием (в том числе наследуемые), длина псевдонима превышает возможную, имя зарезервировано.

Пример:

```
CREATE VIRTUAL TYPE new_virtual_type AS SELECT * FROM old_table_1
```

DBCommand

3.3. ALTER TYPE (AA)

ALTER TYPE - Изменяет свойства типа, добавляет/удаляет атрибуты, создаёт первичные ключи.

Синтаксис

```
ALTER TYPE type_name
{
    SET
        DEFAULT ACL = `acl_id` |
        AUDIT { ALL | CUSTOM }

    | ADD
    {
        ATTRIBUTE <список атрибутов> [ , ... n ]
        | ( FOREIGN KEY (<список атрибутов> [ , ... n ])
            REFERENCES referenced_type_name (<список атрибутов> [ , ... n ])
            [ ON UPDATE { CASCADE | NO ACTION | SET NULL | SET DEFAULT_ } ]
            [ ON DELETE { CASCADE | NO ACTION | SET NULL | SET DEFAULT_ } ] )
        | UNIQUE (<список атрибутов> [ , ... n ] )
    }
    | DROP ATTRIBUTE <список атрибутов> [ , ... n ]
} [ ; ]
```

type_name

Имя псевдонима типа данных или определяемого пользователем типа данных;

referenced_type_name

Имя псевдонима типа данных, на который ссылается ограничение FOREIGN KEY.

SET

Изменяет свойства типа;

DEFAULT ACL = `acl_id`

Назначение матрицы доступа;

AUDIT { ALL | CUSTOM } – у указанного типа будут аудироваться все события – ALL, либо выбранные - CUSTOM

ADD

Добавляет атрибуты к тем, что уже есть в типе;

ATTRIBUTE

Список атрибутов для типа;

FOREIGN KEY REFERENCES

Ограничение, которое обеспечивает ссылочную целостность данных в столбце или столбцах;

ON UPDATE { CASCADE | NO ACTION | SET NULL | SET DEFAULT_ }

Указывает, какая операция производится над строками изменяемой таблицы, если эти строки имеют ссылочное отношение, а строка, на которую имеются ссылки, обновляется в родительской таблице. Значение по умолчанию — NO ACTION.

CASCADE

Если из родительской таблицы удаляется строка, из ссылающейся таблицы удаляются соответствующие ей строки.

NO ACTION

При изменении этого типа связанный с ним меняться не будет;

SET NULL

Все значения, составляющие внешний ключ, при изменении соответствующей строки родительской таблицы устанавливаются в NULL.

SET DEFAULT_

Все значения, составляющие внешний ключ, при обновлении соответствующей строки родительской таблицы устанавливаются в значение по умолчанию. Для выполнения этого ограничения все атрибуты внешних ключей должны иметь определения по умолчанию. Если атрибут может принимать значение NULL, а значение по умолчанию явно не задано, неявным значением по умолчанию для данного атрибута становится NULL.

ON DELETE { CASCADE | NO ACTION | SET NULL | SET DEFAULT_ }

Определяет операцию, которая производится над строками создаваемой таблицы, если эти строки имеют ссылочное отношение, а строка, на которую имеются ссылки, удаляется из родительской таблицы. Значение по умолчанию — NO ACTION.

CASCADE

Если из родительской таблицы удаляется строка, из ссылающейся таблицы удаляются соответствующие ей строки.

NO ACTION

При изменении этого типа связанный с ним меняться не будет;

SET NULL

Все значения, составляющие внешний ключ, при изменении соответствующей строки родительской таблицы устанавливаются в NULL.

SET DEFAULT_

Все значения, составляющие внешний ключ, при обновлении соответствующей строки родительской таблицы устанавливаются в значение по умолчанию. Для выполнения этого ограничения все атрибуты внешних ключей должны иметь определения по умолчанию. Если атрибут может принимать значение NULL, а значение по умолчанию явно не задано, неявным значением по умолчанию для данного атрибута становится NULL.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Изменяет записи об атрибутах в SYS_DD_ATTRS, изменяет таблицы типа, создаёт ключи в таблицах.

Ожидаемый результат

True/exception

Возможные ошибки

Нет прав, несуществующий атрибут/внешний ключ.

Пример:

```
ALTER TYPE user_const ADD attribute h_t INTEGER
```

DBCommand

org.corolla.server.datadictionary.management.commands.oddl.AlterType

3.4. ALTER TYPE SET STORE_POLICY

Команда для изменения политики хранения и миграции контента объектов типа.

Синтаксис

```
ALTER TYPE type_name SET STORE_POLICY STORE_ID = '<store_id>'
    [, MIGRATION TO STORE_ID='<migr_store_id>', MIGRATION AFTER = <интервал в
минутах> ]
```

Аргументы

type_name

Имя псевдонима типа данных или определяемого пользователем типа данных.

'store_id'

ID хранилища, в котором будет храниться контент объектов данного типа

'migr_store_id'

ID хранилища, в которое будет перемещаться (мигрировать) контент

интервал_в_минутах

Время, по прошествии которого неиспользованный контент будет мигрирован

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Изменяет соответствующие поля в таблице SYS_DD_TYPE.

Ожидаемый результат

TRUE/exception.

Возможные ошибки

Указанный тип не существует, либо не поддерживает функциональность контента

Пример:

```
ALTER TYPE VFS_OBJECT SET STORE_POLICY STORE_ID='WSP1000000000001E00000000001JK',  
MIGRATE TO STORE_ID='WSP1000000000001E00000000000RS', MIGRATE AFTER = 600
```

DBCommand

org.corolla.server.datadictionary.management.commands.oddl.alter.ReserveId

3.5. DROP TYPE (AA)

DROP TYPE - Удаляет объектный тип данных из текущего репозитория. В дополнение к удалению типа данных удаляются все таблицы из СУБД и представления.

Невозможно удалить объектный тип данных, если имеются подтипы (порожденные объектные типы данных удаляемого типа или экземпляры удаляемого типа).

Синтаксис

```
DROP TYPE type_name
```

type_name

Имя псевдонима типа данных или определяемого пользователем типа данных;

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Удаляет таблицы, удаляет записи о типе и его атрибутах в SYS_DD_TYPE и SYS_DD_ATTRS

Ожидаемый результат

True/exception

Возможные ошибки

Нет прав, у типа есть объекты, у типа есть наследники.

```
DROP TYPE bsf_commons
```

DBCommand

org.corolla.server.datadictionary.management.commands.oddl.DropType

3.6. DROP VIRTUAL TYPE (ART)

DROP VIRTUAL TYPE - Удаляет виртуальный тип данных.

Синтаксис

```
DROP [ VIRTUAL ] TYPE type_name
```

type_name

Имя типа данных или определяемого пользователем типа данных;

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Удаляет таблицы, удаляет записи о типе и его атрибутах в SYS_DD_TYPE
Ожидаемый результат

True/exception

Возможные ошибки

NONE

```
DROP TYPE bsf_virtual_commons
```

DBCommand

org.corolla.server.datadictionary.management.commands.oddl.DropVirtualType

3.7. DESCRIBE TYPE

Возвращает описание типа данных.

Синтаксис

```
DESCRIBE TYPE <type_name>
```

Аргументы

type_name

Имя псевдонима типа данных или определяемого пользователем типа данных.

Необходимые привилегии - Привилегии для возможности выполнения команды.

Нет

Действие – что делает команда.

Возвращает список атрибутов типа.

Ожидаемый результат

TRUE/exception.

Возможные ошибки

Указанный тип не существует, либо не поддерживает функциональность контента

Пример:

```
DESCRIBE TYPE SYS_DD_TYPE
```

DBCommand

org.corolla.server.datadictionary.management.commands.oddl.DescribeType

4. КОМАНДЫ ODML

Команды манипулирования данными.

4.1. CREATE OBJECT (AA)

CREATE OBJECT - Создаёт объекты (экземпляры) типа.

Синтаксис

```
CREATE OBJECT type_name [ SET ] [ <имя атрибута> = <значение> [ , ... n ] ]
```

type_name

Имя типа данных или определяемого пользователем типа данных;

SET

Устанавливает значения для атрибута/атрибутов.

Необходимые привилегии - Привилегии для возможности выполнения команды.

NONE

Действие – что делает команда.

Создаёт записи в таблице (в родительских таблицах, если тип имеет родителей);

Ожидаемый результат

ID созданного объекта

Возможные ошибки

Неверно указан атрибут.

```
CREATE OBJECT test_type SET a1 = 15, a2 = 'system'
```

DBCommand

org.corolla.server.datadictionary.management.commands.odml.CreateObject

4.2. UPDATE OBJECT (AA)

UPDATE OBJECT - Изменяет значение атрибутов у объектов (экземпляров).

Синтаксис

```
UPDATE OBJECT type_name [ <имя атрибута> = <значение> [ , ... n ] ] [ WHERE <предикат> ]
```

type_name

Имя типа данных или определяемого пользователем типа данных.

SET

Устанавливает значения для атрибута/атрибутов.

< предикат >

Любое выражение, результатом которого являются значения TRUE, FALSE или UNKNOWN.

Необходимые привилегии - Привилегии для возможности выполнения команды.

Права доступа, если тип с безопасностью

Действие – что делает команда.

Изменяет записи об объектах в таблицах базы данных;

Ожидаемый результат

Количество изменённых объектов;

Возможные ошибки

Неверный атрибут. Нет разделителя «,», при этом не происходит ошибки.

```
UPDATE OBJECT test_type SET a1=10, a2='yes'
```

DBCommand

4.3. DELETE OBJECT (AA)

DELETE OBJECT - Изменяет значение атрибутов у объектов.

Синтаксис

```
DELETE OBJECT type_name [ WHERE <предикат> ]
```

type_name

Имя типа данных или определяемого пользователем типа данных.

< предикат >

Любое выражение, результатом которого являются значения TRUE, FALSE или UNKNOWN.

Необходимые привилегии - Привилегии для возможности выполнения команды.

Права доступа, если тип с безопасностью

Действие – что делает команда.

Удаляет записи из таблиц базы данных;

Ожидаемый результат

Количество удалённых записей;

Возможные ошибки

Не забыть указать предикат, чтобы не удалить все объекты.

```
DELETE OBJECT test_type where id = 'WSP000000000100000000RS'
```

DBCommand

org.corolla.server.datadictionary.management.commands.odml.DeleteObject

4.4. SELECT (AA)

SELECT – Осуществляет выборку из объектов типов.

Синтаксис

```
SELECT [ ALL | DISTINCT ] [VERSIONS]
{
  < список выборки > [, ... n]
< список выборки > :: =
{
  < список атрибутов > |
  < STAR > |
  [ { COUNT | MAX | MIN | SUM | AVG ( < STAR > | expression ) } ]
}
FROM { < type_source > } [ , ... n ]
<type_source> :: =
{
  <type_name> <join_type> <type_name>
  [ ON <условия поиска> | USING ( <имя атрибута> ) ]
}
<join_type> :: =
{ INNER | CROSS | { LEFT | RIGHT | FULL } [ OUTER ] }
```

```

JOIN <имя типа2>
[ SEARCH DOCUMENT
  [ CONTENT | METAINFO ]
  '<search_string>'
]
[ WHERE <условия поиска>]
[ GROUP BY
  {
  group_by_expression
  [ HAVING <список условий>]
  } [ , ... n ]
]
[ UNION [ALL] | INTERSECT | MINUS ]
[ ORDER BY
  {
  order_by_expression
  [ ASC | DESC ]
  } [ , ... n]
]
}

```

type_source

Указывает тип, представление или источник производного типа с указанием или без указания псевдонима.

type_name

Имя типа.

join_type

Указывает тип операции соединения.

order_by_expression

Указывает атрибут, по которому должна выполняться сортировка. Атрибут сортировки может быть указан с помощью имени или его псевдонима, или неотрицательного целого числа, представляющего позицию имени или псевдонима в списке выбора.

group_by_expression

Выражение, по которому выполняется группировка.

'search_string'

Поисковая строка, задается тело поискового запроса.

expression

Выражение, такое, как имя атрибута, над которым выполняется вычисление:

Имя атрибута. Только имя атрибута используется в выражении.

Имя переменной или параметр.

Имя таблицы и другие.

ALL

Выборка всех элементов.

DISTINCT

Выборка всех элементов, исключая повторение.

<STAR>

Условие выборки = " * "

AVG

Среднее значение в числовом выражении

COUNT

Количество выбранных строк

MAX

Наибольшее значение в выражении

MIN

Наименьшее значение в выражении

SUM

Сумма значений в числовом выражении

SEARCH DOCUMENT

Используется для определения поискового запроса.

CONTENT

Только в контенте.

METAINFO

Только в атрибутах. Если не указан ни один из предыдущих параметров, то по умолчанию используются оба параметра.

WHERE

Используется для определения, какие строки должны быть выбраны или включены в GROUP BY.

GROUP BY

Используется для объединения строк с общими значениями в элементы меньшего набора строк.

HAVING

Используется для определения, какие строки после GROUP BY должны быть выбраны.

UNION

Указывает на то, что несколько результирующих наборов следует объединить и вернуть в виде единого результирующего набора.

INTERSECT

Возвращает все различные значения, входящие в результаты выполнения запросов, указанных как слева, так и справа от оператора INTERSECT.

MINUS

Вычитает запрос.

ORDER BY

Используется для определения, какие атрибуты используются для сортировки результирующего набора данных.

ASC

Указывает, что значения в указанном атрибуте должны сортироваться по возрастанию, от меньших значений к большим значениям.

DESC

Указывает, что значения в указанном атрибуте должны сортироваться по убыванию, от больших значений к меньшим значениям.

INNER

Указывает, что возвращаются все совпадающие пары строк. Несовпадающие строки из обеих таблиц отбрасываются. Если тип соединения не указан, этот тип задается по умолчанию.

FULL [OUTER]

Указывает, что в результирующий набор включаются строки, как из левой, так и из правой таблицы, несоответствующие условиям соединения, а выходные атрибуты, соответствующие оставшейся таблице, устанавливаются в значение NULL. Этим дополняются все строки, обычно возвращаемые при помощи INNER JOIN.

LEFT [OUTER]

Указывает, что все строки из левой таблицы, не соответствующие условиям соединения, включаются в результирующий набор, а выходные атрибуты из оставшейся таблицы устанавливаются в значение NULL в дополнение ко всем строкам, возвращаемым внутренним соединением.

RIGHT [OUTER]

Указывает, что все строки из правой таблицы, не соответствующие условиям соединения, включаются в результирующий набор, а выходные атрибуты, соответствующие оставшейся таблице, устанавливаются в значение NULL в дополнение ко всем строкам, возвращаемым внутренним соединением.

JOIN

Указывает, что данная операция соединения должна произойти между указанными источниками или представлениями таблицы

ON <условие поиска>

Задаёт условие, на котором основывается соединение. Например:

```
SELECT p.ProductID, v.VendorID
FROM Production.Product AS p
JOIN Purchasing.ProductVendor AS v
ON (p.ProductID = v.ProductID);
```

Необходимые привилегии - Привилегии для возможности выполнения команды.

Права доступа, если тип с безопасностью

Действие – что делает команда.

Делает выборку из представлений типа, накладывая ограничения безопасности;

Ожидаемый результат

ResultSet;

Возможные ошибки

Неверные атрибуты.

```
SELECT ID, NAME FROM SYS_DD_TYPE WHERE ROOT_TYPE_NAME = 'MY_TYPE'
```

DBCommand

org.corolla.server.datadictionary.management.commands.odml.SelectCommand

4.4.1. ПРИМЕРЫ ЗАПРОСОВ ПО ПОЛЬЗОВАТЕЛЯМ И ГРУППАМ

Получить все вложенные группы в данной всех уровней:

Синтаксис

```
SELECT LOWER_ID FROM VIRT_GROUPS_GENEALOGY WHERE UPPER_ID = '<group_id>'
```

Получить все вложенные группы непосредственно в данной:

Синтаксис

```
SELECT LOWER_ID FROM VIRT_GROUPS_GENEALOGY WHERE UPPER_ID = '<group_id>' WHERE  
DISTANCE=1
```

Получить все супергруппы данной группы всех уровней

Синтаксис

```
SELECT UPPER_ID FROM VIRT_GROUPS_GENEALOGY WHERE LOWER_ID = '<group_id>'
```

Получить все непосредственные супергруппы данной группы всех уровней

Синтаксис

```
SELECT UPPER_ID FROM VIRT_GROUPS_GENEALOGY WHERE LOWER_ID = '<group_id>' WHERE  
DISTANCE=1
```

Получить всех пользователей из группы и всех ее подгрупп

Синтаксис

```
SELECT USER_ID FROM VIRT_USER_GROUP_CACHE WHERE GROUP_ID = '<id>'
```

Получить всех пользователей из супергрупп

Синтаксис

```
SELECT USER_ID FROM VIRT_USER_GROUP_CACHE WHERE GROUP_ID IN (SELECT UPPER_ID FROM VIRT_GROUPS_GENEALOGY WHERE LOWER_ID = '<group_id>')
```

'group_id'

Идентификатор группы.

'id'

Идентификатор.

LOWER_ID

Идентификатор группы, находящейся ниже по иерархии, и доступной их группы UPPER_ID.

UPPER_ID

Идентификатор группы, находящейся выше по иерархии.

DISTANCE

Длина пути - количество ребер, которые нужно пройти на пути из группы UPPER_ID в группу LOWER_ID

Необходимые привилегии - Привилегии для возможности выполнения команды.

NONE

4.5. GET AUDIT (AA)

GET AUDIT – Возвращает записи аудита.

Синтаксис

```
GET AUDIT WHERE <предикат>
```

type_name

Имя типа данных или определяемого пользователем типа данных.

< предикат >

Любое выражение, результатом которого являются значения TRUE, FALSE или UNKNOWN.

Необходимые привилегии - Привилегии для возможности выполнения команды.

AUDIT_MANAGER

Действие – что делает команда.

Делает выбор из таблицы AUDITTRAILS по указанному предикату.

Ожидаемый результат

SimpleMultiRowResultSetImpl.

Возможные ошибки

Неверный предикат.

```
GET AUDIT WHERE event_name = 'UPDATE_OBJECT'
```

DBCommand

org.corolla.server.datadictionary.management.commands.odml.GetAudit

5. КОМАНДЫ ADM

5.1. CREATE USER

CREATE USER – Создает пользователя.

Синтаксис

```
CREATE USER 'user_login' PASSWORD 'user_password'  
  [ USER_TYPE = user_type ]  
  [ USER_ATTRS ( < имя атрибута > = '< значение >' [, ... n ] ) ]  
  [ GROUP 'group_name'  
    [ GROUP_TYPE = <имя типа группы> ]  
    [ GROUP_ATTRS ( <имя атрибута> = '< значение >' [, ... n ] ) ]
```

'user_login'

Логин пользователя, латинскими буквами.

'user_password'

Пароль пользователя.

user_type

Указывается пользовательский тип.

group_name

Имя группы.

PASSWORD

Обязательный параметр для установки пароля пользователю.

USER_TYPE

Указывается, чему равен пользовательский тип.

USER_ATTRS

Атрибуты пользователя.

GROUP

Группа, в которую включен пользователь.

GROUP_TYPE

Тип группы.

GROUP_ATTRS

Атрибуты группы.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Создает пользователя в системе;

Ожидаемый результат

Созданный пользователь;

Возможные ошибки

Пользователь с таким логином уже существует, не достаточно прав для создания пользователя, ошибка в синтаксисе (не проставлены кавычки ``");

```
CREATE USER `userlogin` PASSWORD `123` USER_ATTRS ( EMAIL = `null@test.ru`,  
FIRST_NAME=`Johan`)
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.CreateComplexUser

5.2. ALTER USER

ALTER USER – Изменяет свойства объекта.

Синтаксис

```
ALTER USER `user_login`  
{  
    SET < имя атрибута > = `< значение >` [, ... n ]  
}
```

`user_login`

Логин пользователя.

SET

Изменяет свойства указанного в запросе атрибута.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Изменяет свойства объекта;

Ожидаемый результат

Измененные атрибуты объекта;

Возможные ошибки

Нет прав на изменение свойств объекта, ошибка в синтаксисе (не проставлены кавычки ``");

```
ALTER USER `userlogin` SET EMAIL =`compas@null.ru`, FIRST_NAME=`Frederic`
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.AlterUser

5.3. DELETE USER ()

DELETE USER – команда на удаление пользователя.

Синтаксис

```
DELTE USER `user_login`
```

'user_login'

Логин пользователя.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Удаляет объект;

Ожидаемый результат

Удаленный объект;

Возможные ошибки

Нет прав на удаление объекта, ошибка в синтаксисе (не проставлены кавычки "`"),
пользователь с таким логином не существует;

```
DELTE USER `Johan`
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.DropUser

5.4. CREATE GROUP (ART)

CREATE GROUP – создает пользовательскую группу.

Синтаксис

```
CREATE GROUP `group_name`  
[ GROUP_TYPE = group_type_name ( <имя атрибута> = '<значение>' [, ... n] ) ]
```

'group_name'

Имя создаваемой группы.

group_type_name

Указывает имя типа группы.

GROUP_TYPE

Указывает используемый тип для хранения информации.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Создает группу;

Ожидаемый результат

Созданная группа;

Возможные ошибки

Нет прав на создание объекта, ошибка в синтаксисе (не проставлены кавычки `` ` `"), группа с таким именем уже существует;

```
CREATE GROUP 'newgroup' GROUP_TYPE = SYS_GROUP_DATA (NAME = 'new_world_group',  
EMAIL = 'null@test.ru')
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.CreateGroup

5.5. ALTER GROUP (ART)

ALTER GROUP – изменяет свойства группы.

Синтаксис

```
ALTER GROUP 'group_name'  
{  
    SET <имя атрибута> = '<значение>' [, ... n]  
}
```

'group_name'

Имя группы.

SET

Изменяет свойства указанного в запросе атрибута.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Вносит изменения в свойства группы;

Ожидаемый результат

True/exception;

Возможные ошибки

Нет прав на изменение объекта, ошибка в синтаксисе (не проставлены кавычки " ` "),
не существует указанных атрибутов;

```
ALTER GROUP `newgroup` SET NAME = `new_world_group_1`, EMAIL = `group@old.ru`
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.AlterGroup

5.6. DELETE GROUP (ART)

DELETE GROUP – удаляет группу.

Синтаксис

```
DELETE GROUP [ ID ] 'group_name'
```

'group_name'

Имя группы.

ID

Идентификатор группы.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Удаляет группу;

Ожидаемый результат

True/exception;

Возможные ошибки

Нет прав на удаление объекта, ошибка в синтаксисе (не проставлены кавычки " ` "),
группа не существует;

```
DELETE GROUP ID 'WSP000033000000000000000001'
```

```
DELETE GROUP 'group_name'
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.DropGroup

5.7. GET USERS FROM GROUP (ART)

GET USERS FROM GROUP – Выбирает пользователей из группы.

Синтаксис

```
GET USERS FROM GROUP 'group_name' [ RECURSIVE ]
```

'group_name'

Имя группы.

RECURSIVE

Если не указан, то выбирается только первый уровень вхождения.

Необходимые привилегии - Привилегии для возможности выполнения команды.

NONE

Действие – что делает команда.

Выбирает пользователей из группы;

Ожидаемый результат

Выбирает данные о пользователях/exception;

Возможные ошибки

Ошибка в синтаксисе (не проставлены кавычки " ` "), группа не существует;

```
GET USERS FROM GROUP `root_grp`
```

DBCommand

```
org.corolla.server.datadictionary.management.commands.adm.usersgroups.SelectUsersFromGroup
```

5.8. ADD USER TO GROUP ()

ADD USER TO GROUP – Включает пользователя в группу.

Синтаксис

```
ADD USER `user_name` TO GROUP `group_name`
```

'user_name'

Имя пользователя, которого включают в группу.

'group_name'

Имя группы.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Включает пользователя в группу;

Ожидаемый результат

True/exception;

Возможные ошибки

Нет прав на включение пользователя в группу, ошибка в синтаксисе (не проставлены кавычки " ` "), группа не существует, пользователя не существует, пользователь заблокирован;

```
ADD USER 'johan' TO GROUP 'unit1_grp'
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.AddUserToGroup

5.9. REMOVE USER FROM GROUP ()

REMOVE USER FROM GROUP – Исключает пользователя из группы.

Синтаксис

```
REMOVE USER 'user_name' FROM GROUP 'group_name'
```

'user_name'

Имя пользователя, которого исключают из группы.

'group_name'

Имя группы.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Исключает пользователя из группы;

Ожидаемый результат

True/exception;

Возможные ошибки

Нет прав на исключение пользователя, ошибка в синтаксисе (не проставлены кавычки " ` "), группа не существует, пользователя не существует, пользователь заблокирован;

```
REMOVE USER 'johan' FROM GROUP 'unit1_grp'
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.RemoveUserFromGroup

5.10. ADD GROUP TO GROUP (ART)

ADD GROUP TO GROUP – Включает группу в группу.

Синтаксис

```
ADD GROUP [ ID ] 'group_name' TO GROUP [ ID ] 'group_name1'
```

'group_name'

Имя первой группы.

'group_name1'

Имя второй группы.

ID

Идентификатор группы.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Включает группу в группу;

Ожидаемый результат

True/exception;

Возможные ошибки

Нет прав, ошибка в синтаксисе (не проставлены кавычки " ` "), группа или идентификатор группы не существует;

```
ADD GROUP ID 'WSP00039999999923' TO GROUP ID 'WSP00000000000000000243'
```

```
ADD GROUP 'group1' FROM GROUP 'group2'
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.AddGroupToGroup

5.11. REMOVE GROUP FROM GROUP (ART)

REMOVE GROUP FROM GROUP – Исключает группу из группы.

Синтаксис

```
REMOVE GROUP [ ID ] 'group_name' FROM GROUP [ ID ] 'group_name1'
```

'group_name'

Имя первой группы.

'group_name1'

Имя второй группы.

ID

Идентификатор группы.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Исключает группу из группы;

Ожидаемый результат

True/exception;

Возможные ошибки

Нет прав, ошибка в синтаксисе (не проставлены кавычки `` ` `”), группа или идентификатор группы не существует;

```
REMOVE GROUP ID 'WSP00039999999923' FROM GROUP ID 'WSP0000000000000000243'
```

```
REMOVE GROUP 'group1' FROM GROUP 'group2'
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.RemoveGroup
FromGroup

5.12. RECALCULATE CACHE

RECALCULATE CACHE – Пересчитать кэш групп и пользователей.

Синтаксис

```
RECALCULATE CACHE
```

Необходимые привилегии - Привилегии для возможности выполнения команды.

NONE

Действие – что делает команда.

Пересчитывает кэш групп и пользователей;

Ожидаемый результат

true/exception;

Возможные ошибки

Нет;

RECALCULATE CACHE

DBCommand

org.corolla.server.datadictionary.management.commands.adm.usersgroups.RecalculateCache

5.13. PURGE AUDIT

PURGE AUDIT – Удаление аудита.

Синтаксис

```
PURGE AUDIT
  [ EVENT 'event_name' ]
  [ UNTIL <кол-во часов> HOURS | TO_DATE ( 'date', 'date_format' ) ]
```

'group_name'

Имя группы.

'event_name'

Имя события.

'date'

Дата

'date_format'

Формат даты.

HOURS

Указывается количество часов.

TO_DATE

Указывается дата и формат даты.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Удаляет весь аудит или только за указанный период;

Ожидаемый результат

true/exception;

Возможные ошибки

Нет прав. Ошибка в синтаксисе (не проставлены кавычки " ` "), неправильный формат даты;

```
PURGE AUDIT 'read' TO_DATE ('10.08.2008', 'dd.MM.YYYY')
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.audit.PurgeAudit

5.14. PURGE STATISTICS

PURGE STATISTICS – Удаление статистики.

Синтаксис

```
PURGE STATISTICS <имя типа или атрибута>  
[ UNTIL <кол-во часов> HOURS | TO_DATE ( 'date', 'date_format' ) ]
```

'group_name'

Имя группы.

'date'

Дата

'date_format'

Формат даты.

HOURS

Указывается количество часов.

TO_DATE

Указывается дата и формат даты.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Удаляет всю статистику за всё время или за указанный период;

Ожидаемый результат

true/exception;

Возможные ошибки

Нет прав. Ошибка в синтаксисе (не проставлены кавычки " ` "), неправильный формат даты;

```
PURGE STATISTIC SYS_GROUP_DATA TO_DATE ('10.08.2008', 'dd.MM.YYYY')
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.stat.PurgeStatistics

5.15. EXECUTE SQL ()

EXECUTE SQL – Выполнение SQL запроса напрямую.

Синтаксис

```
EXECUTE SQL 'query'
```

'query'

Строка SQL запроса.

Необходимые привилегии - Привилегии для возможности выполнения команды.

DD_MANAGER

Действие – что делает команда.

Удаляет всю статистику за всё время или за указанный период;

Ожидаемый результат

true/exception;

Возможные ошибки

Нет прав. Ошибка в синтаксисе (не проставлены кавычки " ` "), неправильный формат даты;

```
EXECUTE SQL 'SELECT * FROM SYS_GROUP_DATA'
```

DBCommand

org.corolla.server.datadictionary.management.commands.adm.ExecuteSql

6. КОМАНДЫ PREPARED STATEMENT