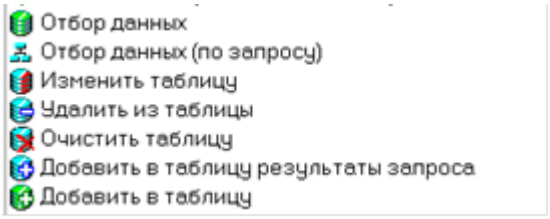
Работа с таблицами БД

Для работы с таблицами БД применяется SQL ((Structured Query Language) - структурный язык запросов. С основами SQL можно ознакомиться в Приложении ‘SQL в Домино’.

Далее рассмотрим операторы для работы с таблицами:

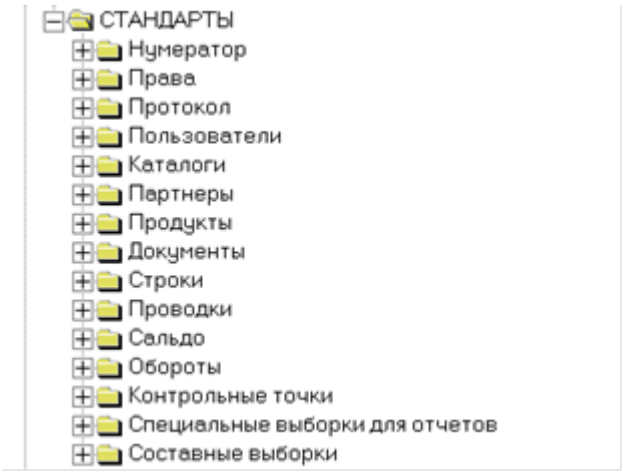


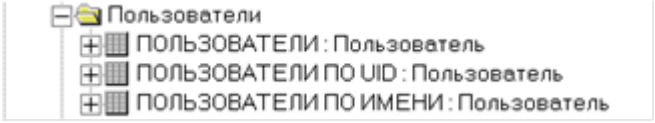
Оператор Отбор данных

Оператор ‘*Отбор данных*’ применяется для последовательного считывания записей из основных таблиц БД с применением стандартных индексов. Стандартные индексы описаны в проекте особым образом.

> Данная конструкция применяется редко, поскольку имеется более удобная конструкция ‘Отбор данных по запросу’ с дополнительными атрибутами.

Индексы основных таблиц БД формируются в момент создания БД. В проекте имеется лишь общее описание стандартных индексов - в библиотеке ‘*Системная область*’, раздел ‘*Выборки* ’, папка ‘*Стандарты*’.

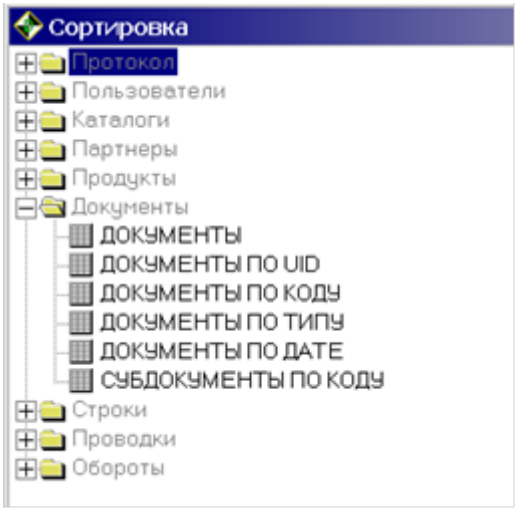




Стандартные индексы таблицы Пользователи.

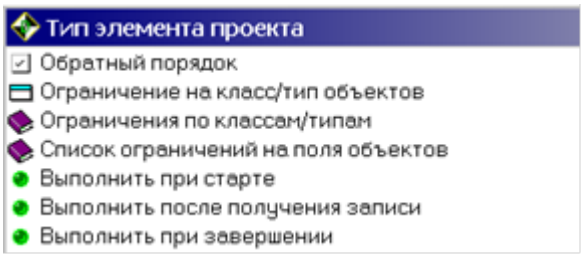
Применение стандартных индексов предпочтительно по той причине, что отбор данных будет происходить максимально быстро.

При использовании оператора ‘*Отбор данных*’ необходимо указать стандартный индекс, который предполагается применить для считывания данных из таблицы. Для выбора отображаются все ранее перечисленные индексы.



Окно для выбора стандартного индекса.

После выбора стандартного индекса можно заполнить остальные атрибуты оператора:

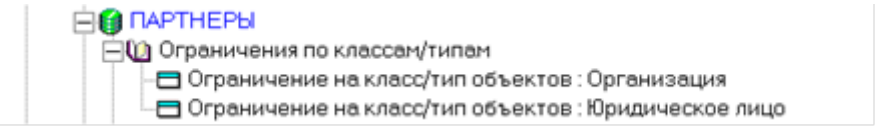


- **Обратный порядок** – при установленном признаке применяется обратный (реверсный) порядок чтения записей индекса. Все стандартные индексы, кроме индексов типа ‘*** ПО ДАТЕ’ имеют сортировку по возрастанию значений. Индексы типа ‘*** ПО ДАТЕ’ имеют сортировку по убыванию значения даты, т.е. записи располагаются от поздних дат к ранним.
- **Ограничения на класс/тип объектов** – применяется для задания одного ограничения на класс или тип выбираемых записей. Можно указать либо проектный элемент, либо выражение, возвращающее UID нужного проектного элемента.

Интересуют партнеры только класса ‘Подразделение’.

Ограничения по классам/типам – применяется для задания нескольких ограничений на

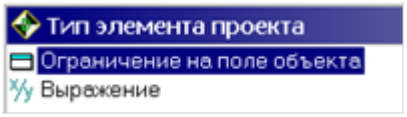
классы и типы выбираемых записей. В остальном описание аналогично предыдущему атрибуту.



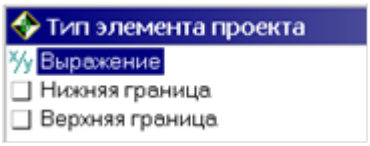
Интересуют партнеры только типов ‘Организация’ и ‘Юридическое лицо’.

Задание ограничений на класс и тип вынесены в отдельный атрибут поскольку поля ‘Класс’ и ‘Тип’ имеются во многих индексах. Учитывая эту особенность, программа пытается сконструировать более эффективный запрос к БД.

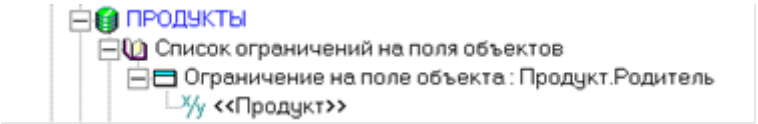
- **Список ограничений на поля объектов** - применяется для задания ограничений на остальные поля таблицы.



Для каждого поля можно указать следующие условия:



  - **Выражение** – задает условие равенства. Если необходимо указать набор допустимых значений, то следует указать несколько выражений.



Интересуют наследники продукта, указанного в контекстной переменной <<Продукт>>.

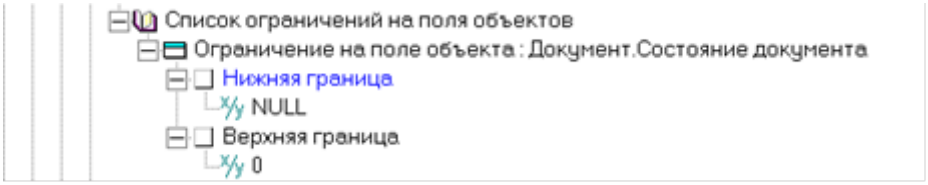
  - **Нижняя граница, Верхняя граница** – применяются для задания диапазона значений. Границы включаются в диапазон. Если указать значение NULL, то это означает, что соответствующая граница отсутствует.

При задании ограничений по дате применяются дополнительные правила:

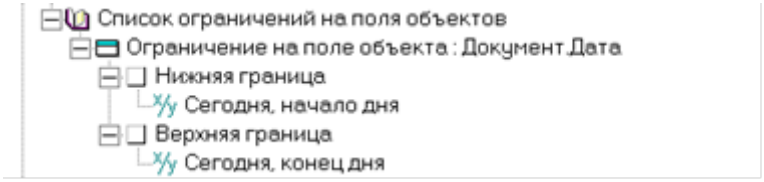
    - Если в качестве верхней границы указана дата со временем **равным 0**, то к верхней границе добавляются сутки;
    - Если в качестве верхней границы указана дата со

временем **не равным 0**, то к верхней границе добавляется одна секунда.
- **В обоих случаях** верхняя граница в диапазон не включается.

Данные правила введены для удобства задания ограничений по дате.



Интересуют только неакцептованные документы.



Интересуют все документы за текущий день.

- **Выполнить при старте -** Последовательность операторов, которые будут выполнены сразу после получения первой записи. Если данных нет, то блок '*Выполнить при старте*' не вызывается.
- **Выполнить после получения записи - О**ператоры, которые будут выполнены после получения каждой следующей записи из таблицы. Если больше записей нет, то блок '*Выполнить после получения записи*' не вызывается.

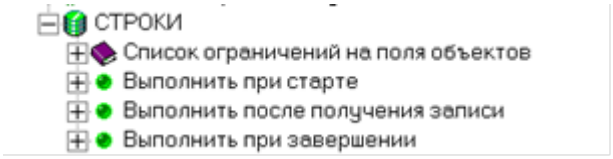
  При получении **первой** записи блок '*Выполнить после получения записи*' вызывается **после** вызова блока '*Выполнить при старте*'. При получении **последней** записи блок '*Выполнить после получения записи*' вызывается **перед** блоком '*Выполнить при завершении*'.

- **Выполнить при завершении** - Операторы, которые будут выполнены после получения последней записи. Если записей не было, то блок '*Выполнить при завершении*' не вызывается.

Одновременное указание всех блоков не обязательно, и определяется логикой программы.

После завершения считывания записей из таблицы управление возвращается в процедуру.

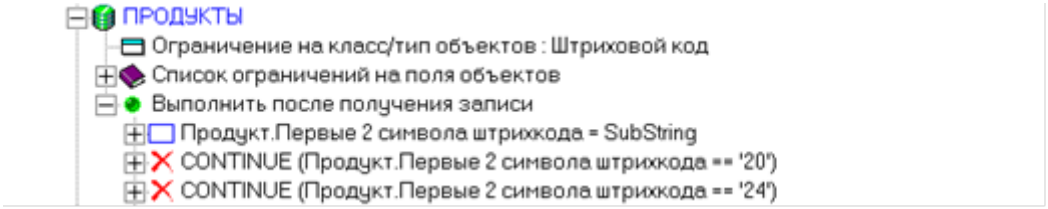
Пример.



Для данной таблицы указано максимально возможное количество вызовов блоков-событий.

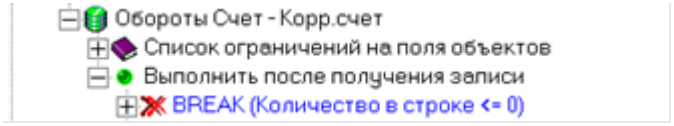
При отборе данных можно использовать операторы ‘CONTINUE’, ‘BREAK’ и ‘*Завершить блок действий*’.

Оператор ‘CONTINUE’ вызывает переход к получению следующей записи. Если оператор ‘CONTINUE’ вызвать из блока ‘*Выполнить при старте*’, то выполнение этого блока прервется.



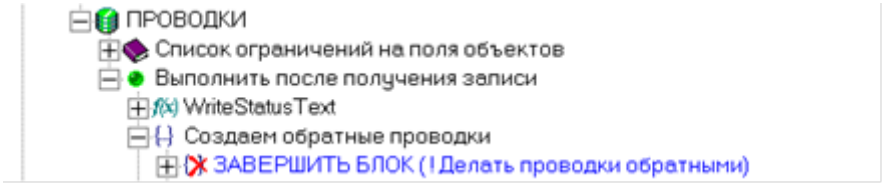
Если первые два символа штрихкода товара равны 20 или 24, то обработка текущего блока прерывается, и считывается следующая запись.

Оператор ‘BREAK’ прерывает отбор данных. Блок ‘*Выполнить при завер*шении’ не вызывается.



Если для полученной строки выполняется условие оператора BREAK, то считывание записей из таблицы прерывается. Блок ‘Выполнить при завершении’ отсутствует.

Оператор ‘*Завершить блок действий*’ завершает выполнение текущего блока-события. Отбор данных не прерывается, и последовательность вызова очередных блоков-событий не меняется.

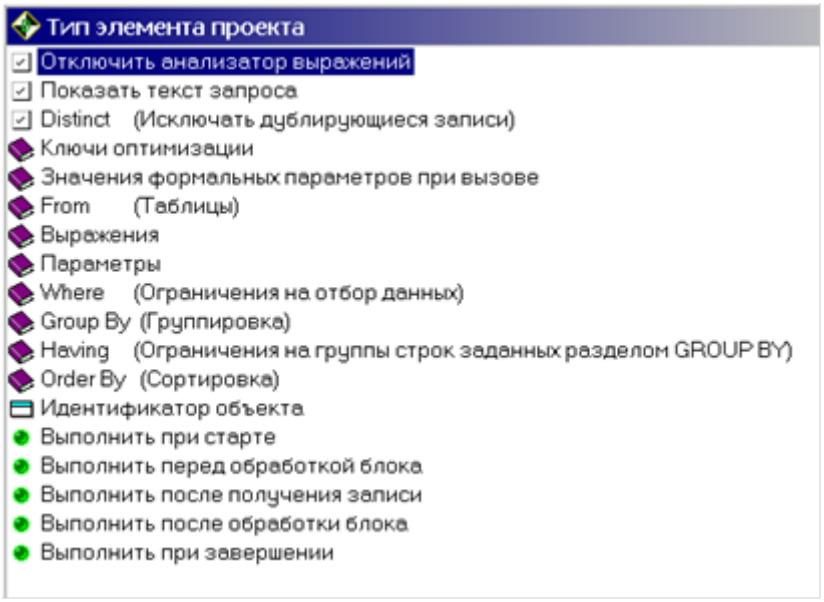


Если не разрешено создавать обратные проводки, то обработка текущего блока прерывается

Оператор Отбор данных (по запросу)

Оператор ‘*Отбор данных (по запросу)*’ применяется для последовательного считывания записей из основных таблиц БД по указанному запросу.

Можно описать новый запрос, можно сослаться на существующий запрос. Основные правила описания запросов были рассмотрены выше. Рассмотрим назначение остальных атрибутов.



- **Показать текст запроса** – применяется для отладки запроса. При установленном признаке сформированный на языке SQL запрос будет отображен на экране.
- **Идентификатор объекта** – ссылка на параметр запроса, значение которого определяет текущий объект. По умолчанию, запрос не выставляет текущий объект. Использование данного параметра позволяет внутри отбора данных применять функции, работающие от текущего объекта.
- **Выполнить при старте -** Последовательность операторов, которые будут выполнены сразу после получения первой записи. Если данных нет, то блок ‘*Выполнить при старте* ’ не вызывается.
- **Выполнить перед обработкой блока - О**ператоры, которые будут выполнены после получения следующей записи, если изменились значения описанных в блоке ключевых полей. При получении первой записи блок ‘*Выполнить перед обработкой блока*’ вызывается сразу после вызова блока ‘*Выполнить при старте*’.
- **Выполнить после получения записи - О**ператоры, которые будут выполнены после получения каждой следующей записи из таблицы. Если больше записей нет, то блок ‘ *Выполнить после получения записи*’ не вызывается.

  При получении **первой** записи блок ‘*Выполнить после получения записи*’ вызывается **после** вызова блока ‘*Выполнить при старте*’ и всех блоков ‘*Выполнить перед обработкой блока*’. При получении **последней** записи блок ‘*Выполнить после получения записи*’ вызывается **перед** вызовом всех блоков ‘*Выполнить после обработки блока*’ и блока ‘*Выполнить при завершении*’.

- **Выполнить после обработки блока - О**ператоры, которые будут выполнены после

получения последней записи с текущими значениями описанных в блоке ключевых полей.

При получении последней записи блок ‘*Выполнить после обработки блока*’ вызывается перед вызовом блока ‘*Выполнить при завершении’*.

- **Выполнить при завершении** - Операторы, которые будут выполнены после получения последней записи. Если записей не было, то блок ‘*Выполнить при завершении*’ не вызывается.

Одновременное указание всех блоков не обязательно, и определяется логикой программы.

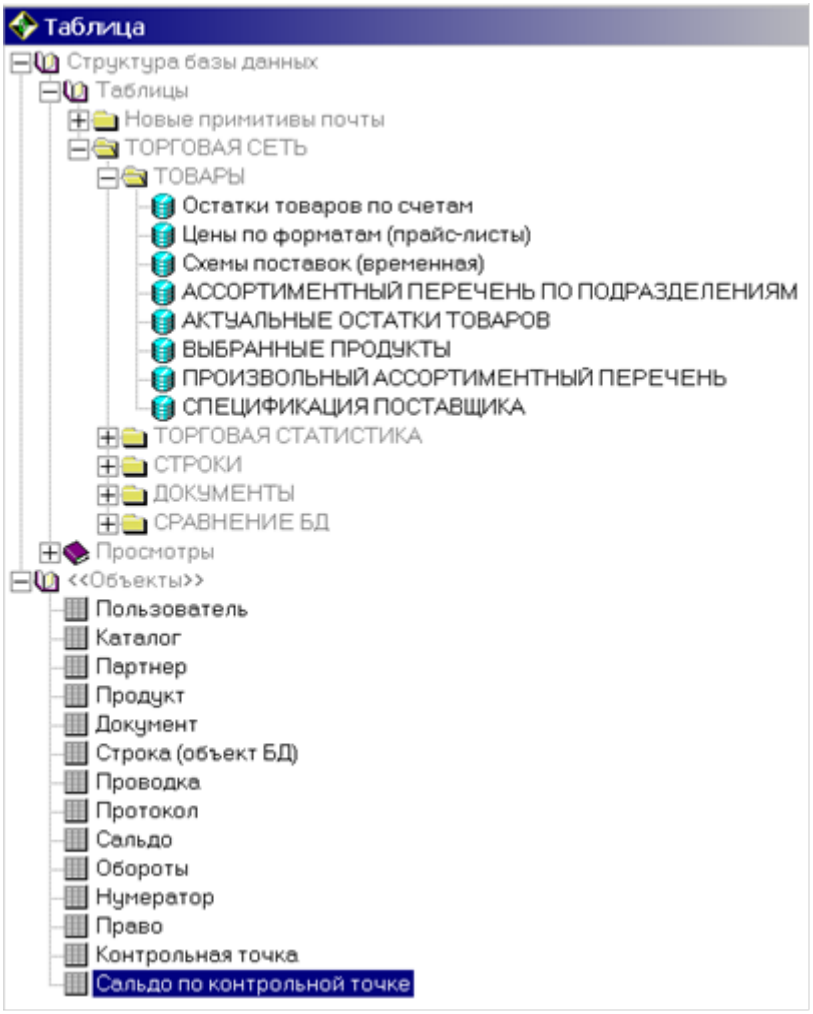
После завершения считывания записей из таблицы управление возвращается в процедуру.

При отборе данных можно использовать операторы ‘CONTINUE’, ‘BREAK’ и ‘*Завершить блок действий*’. Действие данных операторов в конструкции ‘*Отбор данных (по запросу)*’ аналогично их действию в конструкции ‘*Отбор данных*’.

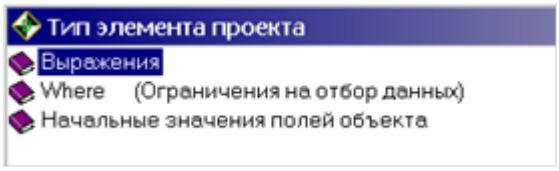
Оператор Изменить таблицу

Данный оператор изменяет указанные записи таблицы.

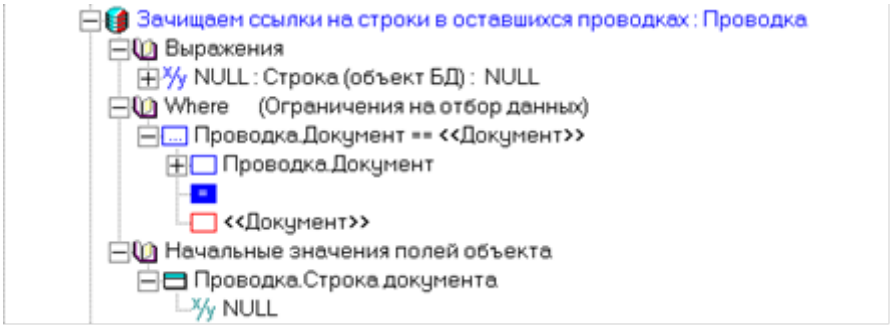
При использовании оператора ‘*Изменить таблицу*’ необходимо указать либо одну из основных таблиц БД (в списке они расположены в папке <<Объекты>>), либо любую из дополнительных таблиц.



Для оператора можно указать следующие атрибуты:



- **Выражение** – выражения из этой папки применяются при описании других атрибутов.
- **WHERE (Ограничение на отбор данных)** – задает условие для отбора изменяемых записей.
- **Начальные значения полей объекта** – список полей с новыми значениями.



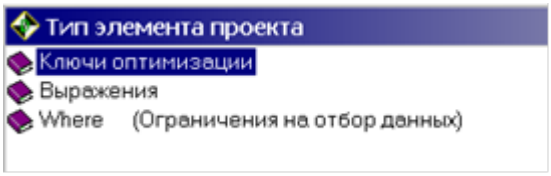
Из таблицы ‘Проводки’ отбираются все записи, связанные с интересующим документом (UID документа находится в контекстной переменной <<Документ>>). Для каждой полученной строки очищается поле ‘Строка документа’.

Оператор Удалить из таблицы

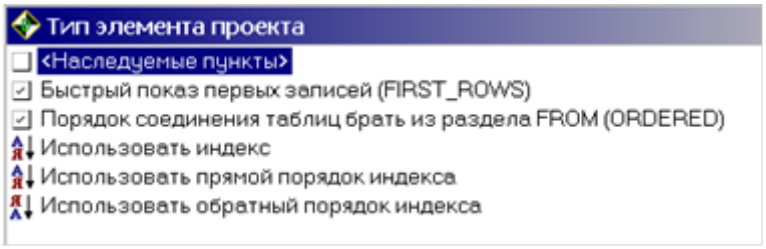
Данный оператор удаляет записи из таблицы.

При использовании оператора ‘*Удалить из таблицы*’ необходимо указать либо одну из основных таблиц БД (в списке они расположены в папке <<Объекты>>), либо любую из дополнительных таблиц. Список доступных таблиц такой же, как и для оператора ‘ *Изменить таблицу*’.

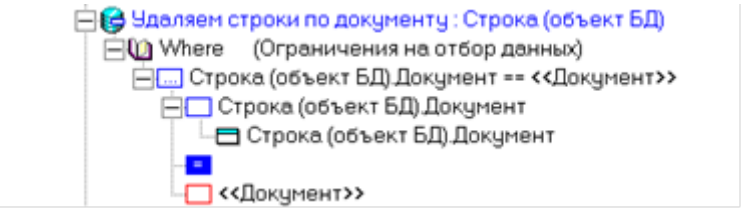
Для оператора можно указать следующие атрибуты:



- **Ключи оптимизации** – Данный раздел содержит инструкции, позволяющие улучшить выполнение запроса.



  - **<Наследуемые пункты>** - этот атрибут применяется для наследования атрибутов раздела.
  - **Быстрый показ первых записей (FIRST_ROWS)** – для оператора удаления записей данный ключ оптимизации не используется.
  - **Порядок соединения таблиц брать из раздела FROM (ORDERED) -** для оператора удаления записей данный ключ оптимизации не используется.
  - **Использовать индекс,**
  - **Использовать прямой порядок индекса,**
  - **Использовать обратный порядок индекса –** позволяют явно указать индекс, который следует применить для эффективного выполнения запроса. Использование индекса может существенно ускорить фильтрацию обрабатываемых данных.
- **Выражение** – выражения из этой папки применяются при описании других атрибутов.
- **WHERE (Ограничение на отбор данных)** – задает условие для отбора удаляемых записей.

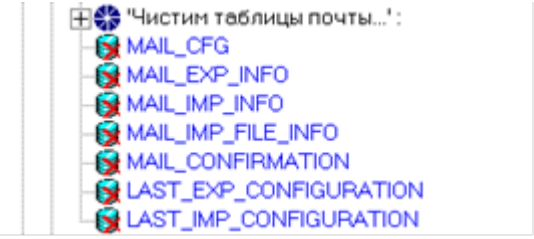


Удаляются все строки документа, заданного в контекстной переменной <<Документ>>.

Оператор Очистить таблицу

Оператор '*Очистить таблицу*' удаляет все записи из таблицы.

При использовании оператора '*Очистить таблицу*' необходимо указать либо одну из основных таблиц БД (в списке они расположены в папке <<Объекты>>), либо любую из дополнительных таблиц. Список доступных таблиц такой же, как и для оператора '*Изменить таблицу*'.



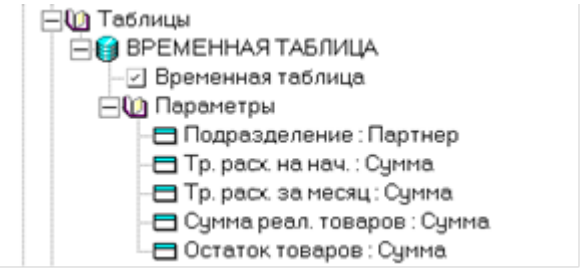
Результаты работы данного оператора не восстанавливаются с откатом транзакции.

Оператор Добавить в таблицу

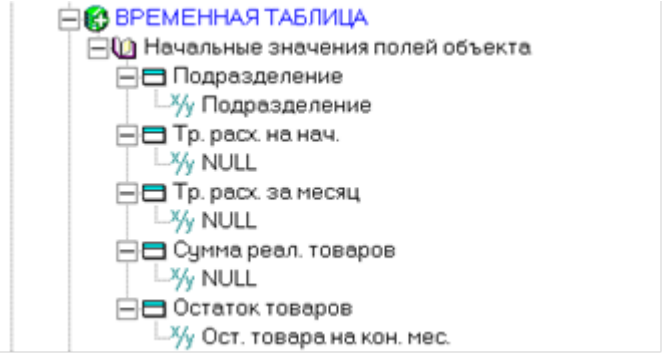
Данный оператор добавляет запись в таблицу.

При использовании оператора '*Добавить в таблицу*' необходимо указать либо одну из основных таблиц БД (в списке они расположены в папке <<Объекты>>), либо любую из дополнительных таблиц. Список доступных таблиц такой же, как и для оператора '*Изменить таблицу*'.

Значения полей указываются в папке '*Начальные значения полей объекта*'.



Описание таблицы

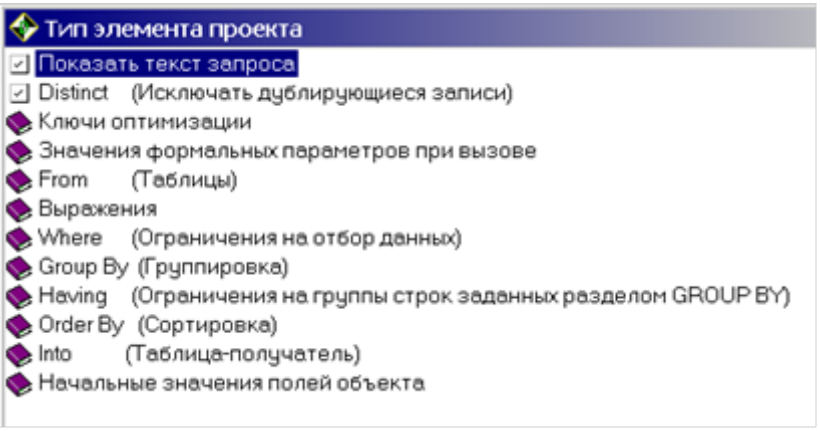


Добавление новой записи в таблицу

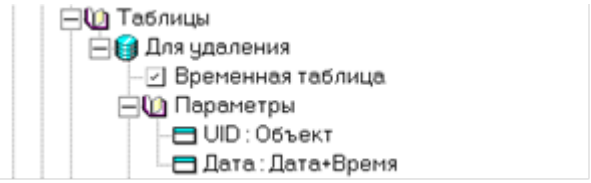
Оператор Добавить в таблицу результаты запроса

Данный оператор добавляет в таблицу несколько записей, полученных в результате выполнения указанного запроса.

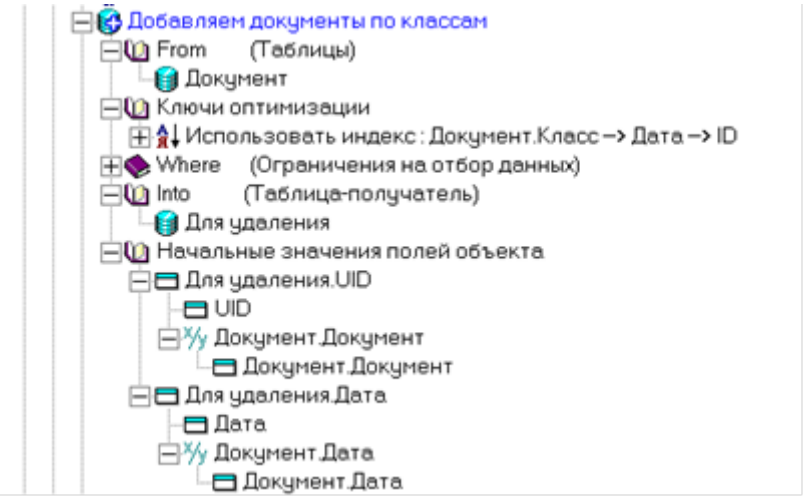
Можно описать новый запрос, можно сослаться на существующий запрос. Основные правила описания запросов были рассмотрены выше. Рассмотрим назначение остальных атрибутов.



- **INTO (Таблица-получатель)** – предназначен для задания таблицы, в которую будут добавлены записи.
- **Начальные значения полей объекта** – содержит список полей таблицы. Для каждого поля можно указать выражение для расчета значения.



Описание таблицы



Результат запроса по таблице ‘Документ’ добавляется в таблицу ‘Для удаления’. Таблица ‘Для удаления’ содержит два столбца. Оба этих столбца заполняются из соответствующих параметров каждой отобранной записи таблицы ‘Документ’.

Версия #1

Демонов Сергей создал Fri, Apr 1, 2022 6:08 AM

Демонов Сергей обновил Fri, Apr 1, 2022 6:25 AM