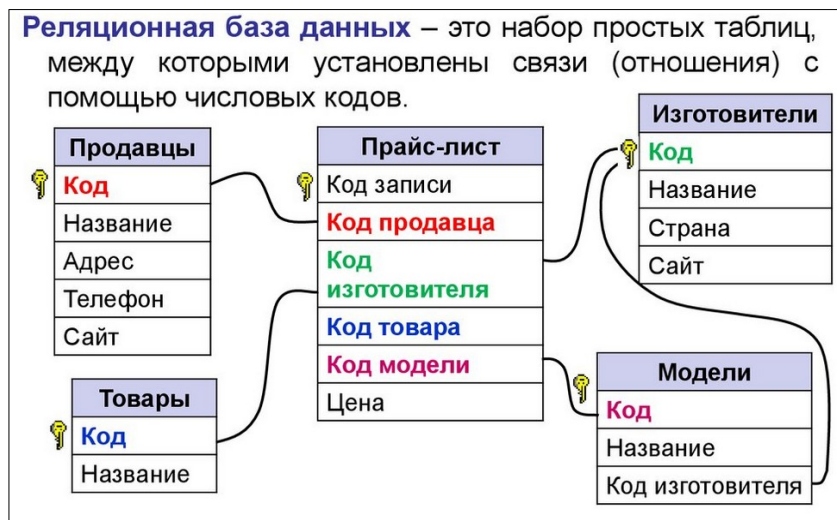


Механизмы запросов 1С

Конфигурации «1С:Предприятие», обрабатывают с данные, которые вводит пользователь, заполняя интерфейсные формы. Из этих форм данные с помощью механизмов платформы записываются в базу данных и хранятся там с тем, чтобы в последствии пользователи могли обратиться к ним для получения итоговой аналитики в том или ином представлении (например, отчетов или диаграмм). Это возможно через использование механизмов запросов, встроенных в технологическую платформу.

Особенности хранения данных прикладных решений в информационной базе

Для хранения данных прикладных решений технологическая платформа использует реляционные базы данных. Реляционная база данных представляет собой совокупность двумерных таблиц, состоящих из набора строк-записей-кортежей и столбцов-полей-колонок.



Ссылочные и нессылочные типы данных

Среди типов данных, поддерживаемых платформой присутствуют так называемые ссылочные типы. К ним можно отнести справочники, документы, планы видов характеристик и т.д. Данные с такими типами содержат *Ссылки* - указатели на соответствующие объекты. Так, например, **объект элемент справочника Сотрудники** имеет поле *Ссылка*, которая будет содержать указатель на этот объект. В дальнейшем этот указатель можно использовать при работе с прочими объектами, которые должны ссылаться на данный элемент справочника и будут иметь соответствующий реквизит с типом данных СправочникСсылка.Сотрудники.

К нессылочным типам можно отнести регистры, которые представляют из себя строки определенной структуры. В некоторых полях будут содержаться значения измерений, а в других ресурсов. К строке регистра невозможно обратиться по ссылке, как к документу или элементу справочника. Для получения конкретной строки регистра используются *Отборы*.

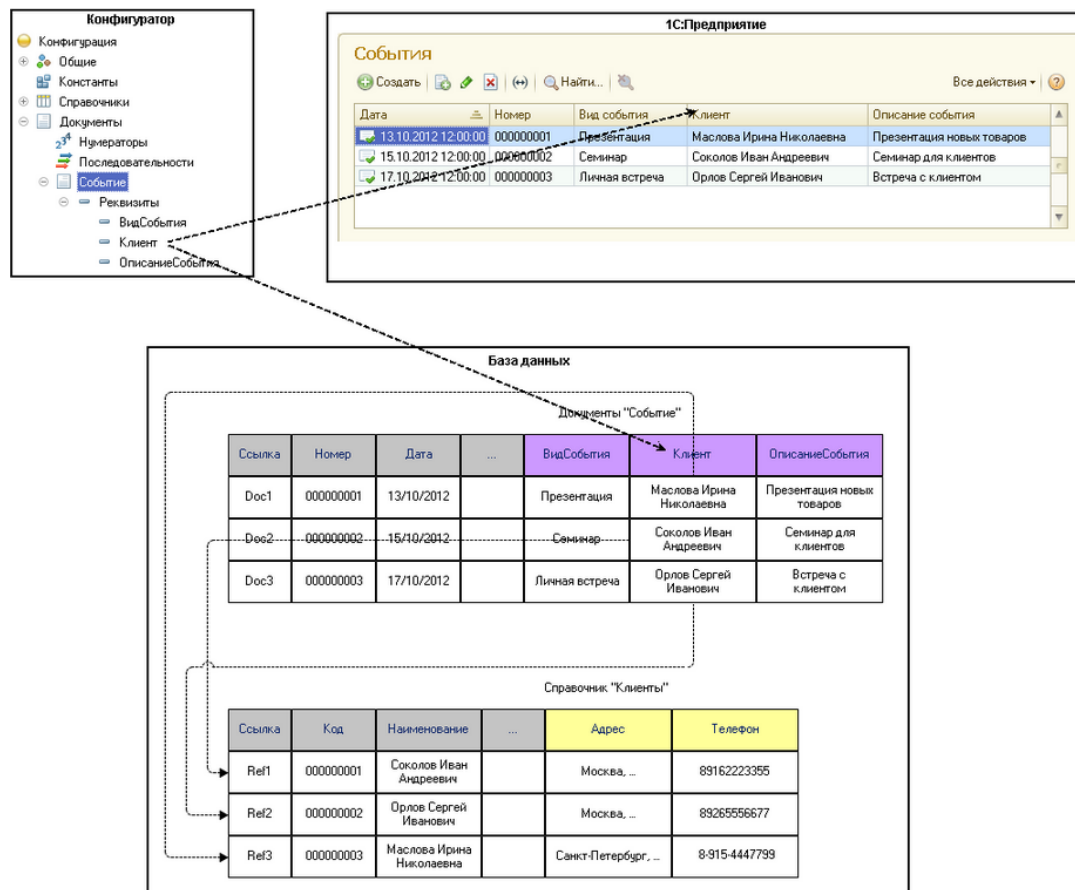
Если удалить объект ссылочного типа, а потом создать новый с абсолютной такими же данными, то с точки зрения платформы это будет уже совершенно другой объект с другим уникальным идентификатором-ссылкой. Если же удалить данные нессылочного типа, например, запись регистра, а потом создать новую с такими же данными, то с точки зрения платформы новая запись ничем не будет отличаться от удаленной, т.е. будет тем же самым.

Хранение данных ссылочных типов в информационной базе

Рассмотрим особенности хранения ссылочных типов данных на примере хранения справочников в информационной базе. Очень часто такие типы данных служат для связи нескольких объектов конфигурации в базе данных.

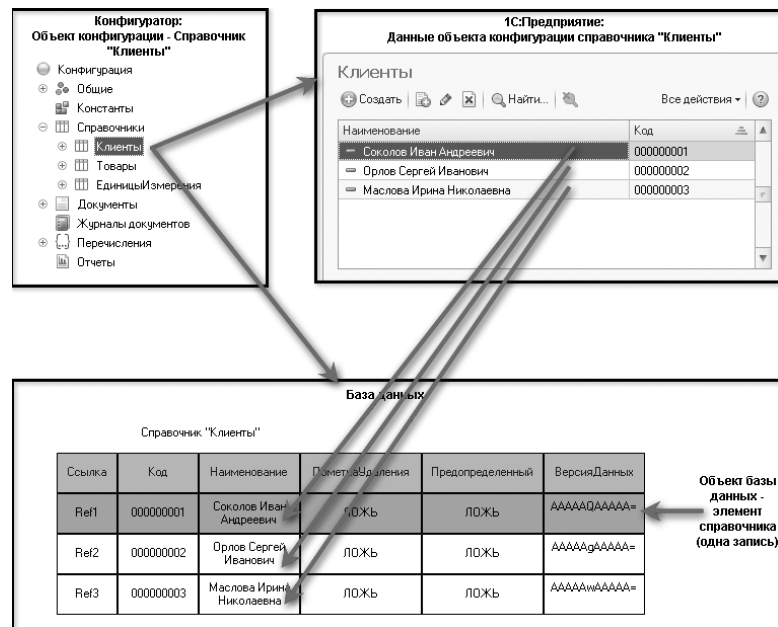
Следует отметить, что конкретные ссылочные типы данных (СправочникСсылка.Клиенты, ДокументСсылка.ПриходнаяНакладная и т.д.) в отличие от примитивных типов данных (число, строка, булево, дата) в конфигурации заранее не определены, а лишь появляются при создании конкретного объекта конфигурации.

Так, на рисунке ниже видно, что документ *Событие* имеет реквизит *Клиент*, который не смотря на свое текстовое представление в визуальном интерфейсе формы, хранит ссылку на запись справочника *Клиенты*, т.к. имеет тип данных *СправочникСсылка.Клиенты*. Работая с документом *Событие*, можно получить всю информацию о выбранном клиенте, обратившись к значениям поля *Клиент* этого документа.



При создании в конфигурации справочника **Клиенты** в информационной базе автоматически создается **основная таблица** этого справочника с полями **Ссылка**, **Код**, **Наименование**, **ПометкаУдаления**, **Предопределенный** и **ВерсияДанных**.

Пользователь заполняет справочник и данные сохраняются в этой основной таблице. Поле **Ссылка** — уникальный идентификатор записи, который создается платформой. Поля **Код**, **Наименование**, **ПометкаУдаления**, **Предопределенный** и **ВерсияДанных** - стандартные реквизиты любого справочника.



Из рисунка видно, что в конфигурации создается **объект метаданных** справочник Клиенты. Другой вид объекта, связанный со справочником Клиенты, **элемент справочника** Клиенты вводится в режиме 1С:Предприятие, после чего его данные записываются в **основную таблицу** справочника Клиенты в базе данных. В таблице же каждому элементу справочника соответствует **одна запись**, однозначно определяющаяся значением поля **Ссылка**.

При добавлении во время разработки реквизитов справочника платформа создает поля соответствующего типа в основной таблице справочника.

У иерархических справочников в структуре основной таблицы обязательно присутствует поле **Родитель** и в случае типа иерархии *Иерархия групп и элементов* будет присутствовать поле **ЭтоГруппа**.

1С:Предприятие

Товары

Создать | [иконки] | Найти... | Все действия

Наименование	Код	Производитель
Товары		
Обувь	000000001	
Детская обувь	000000009	
Пляетки	000000010	
Кроссовки	000000004	
Сапоги	000000003	Италия
Туфли	000000002	Германия
Продукты	000000005	
Масло	000000006	Россия
Молоко	000000007	Россия
Сметана	000000008	

База данных

Иерархический справочник: "Товары" (с иерархией групп и элементов)

Ссылка	Код	Наименование	Родитель	ЭтоГруппа	Производитель
Ref1	000000001	Обувь		ИСТИНА	
Ref9	000000009	Детская обувь	Обувь	ИСТИНА	
Ref10	000000010	Пляетки	Детская обувь	ЛОЖЬ	
Ref4	000000004	Кроссовки	Обувь	ЛОЖЬ	
Ref3	000000003	Сапоги	Обувь	ЛОЖЬ	Италия
Ref2	000000002	Туфли	Обувь	ЛОЖЬ	Германия
Ref5	000000005	Продукты		ИСТИНА	
Ref2	000000006	Масло	Продукты	ЛОЖЬ	Россия
Ref3	000000007	Молоко	Продукты	ЛОЖЬ	Россия
Ref4	000000008	Сметана	Продукты	ЛОЖЬ	

Если справочник является подчиненным, то в его основную таблицу добавляется поле **Владелец**, которое содержит ссылку на элемент справочника-владельца.

1С:Предприятие

Расчетные счета

Создать | [иконки] | Найти... | Все действия

Наименование	Код	Владелец
Счет в СберБанке	000000001	Животноводство ООО
Счет во ВнешТоргБанке	000000002	Корнет ЗАО
Счет в СтройКомБанке	000000003	Скороход АО
Счет в СберБанке	000000004	Скороход АО
Счет в СтройКомБанке	000000005	Корнет ЗАО

Поставщики

Создать | [иконки] | Найти... | Все действия

Наименование	Код
Скороход АО	000000001
Корнет ЗАО	000000002
Животноводство ООО	000000003

База данных

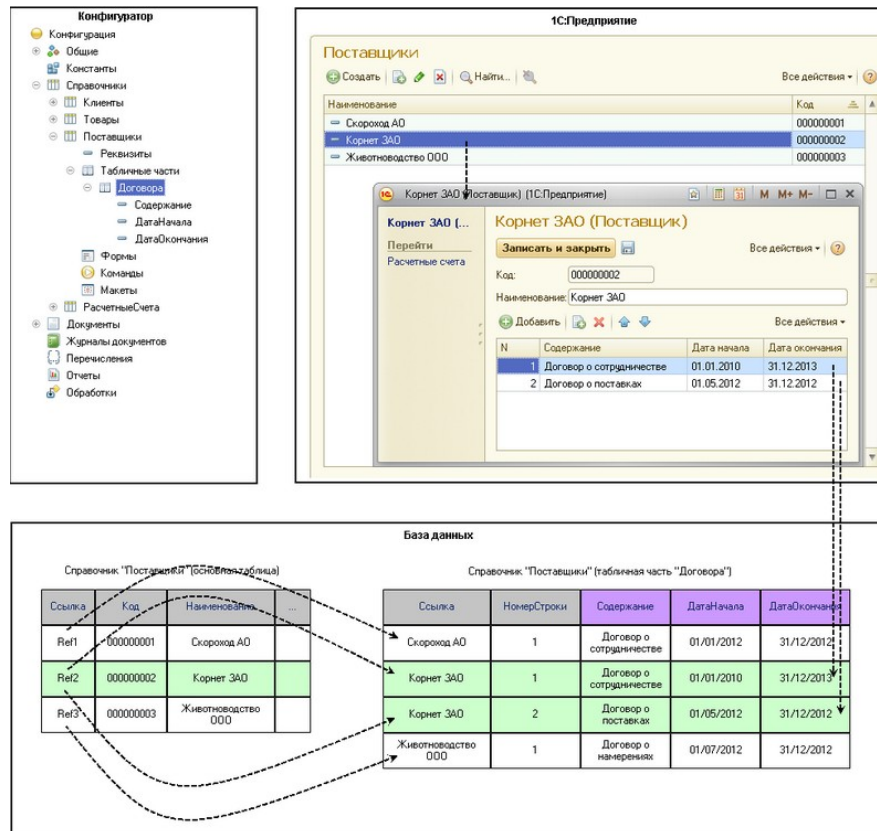
Подчиненный справочник: "РасчетныеСчета"

Ссылка	Код	Наименование	Владелец
Ref1	000000001	Счет в СберБанке	Животноводство ООО
Ref2	000000002	Счет во ВнешТоргБанке	Корнет ЗАО
Ref3	000000003	Счет в СтройКомБанке	Скороход АО
Ref4	000000004	Счет в СберБанке	Скороход АО
Ref5	000000005	Счет в СтройКомБанке	Корнет ЗАО

Справочник-владелец "Поставщики"

Ссылка	Код	Наименование
Ref1	000000001	Скороход АО
Ref2	000000002	Корнет ЗАО
Ref3	000000003	Животноводство ООО

Если у справочника присутствуют табличные части, то в информационной базе же создается *подчиненная таблица* с полями Ссылка и НомерСтроки и полями, соответствующими реквизитами табличной части. Между *основной таблицей* справочника и *подчиненной таблицей* табличных частей существует связь по полю Ссылка. С помощью запросов можно получить всю информацию из табличной части, относящуюся к конкретному элементу справочника. В информационной базе создается столько подчиненных таблиц, сколько табличных частей у справочника.



На примере справочника, имеющего табличную часть, видно, что одному объекту конфигурации в информационной базе могут соответствовать несколько таблиц – одна *основная* и одна или несколько *подчиненных* основной по полю Ссылка. При этом одному объекту *элемент справочника* соответствует одна запись в основной таблице и одна или несколько записей в подчиненных таблицах.

Хранение данных нессылочных типов

Любой регистр предназначен для хранения данных (ресурсов) в разрезе аналитики измерений. Для соответствующего регистра платформа создает в информационной базе таблицу, в которой может храниться произвольная информация, «привязанная» к набору измерений.

Конфигуратор

- Конфигурация
- Общие
- Константы
- Справочники
- Документы
- Журналы документов
- Перечисления
- Отчеты
- Обработки
- Планы видов характеристик
- Планы счетов
- Планы видов расчета
- Регистры сведений
 - Цены
 - Измерения
 - Товар
 - Ресурсы
 - Цена
 - Результаты

1С:Предприятие

Цены

Создать | [иконки] | Найти...

Период	Товар	Цена
15.10.2012	Туфли	7 000
15.10.2012	Сапоги	10 000
15.10.2012	Кроссовки	3 000
20.10.2012	Масло	70
20.10.2012	Молоко	40
20.10.2012	Сметана	50
01.11.2012	Туфли	8 000
01.11.2012	Масло	80
05.11.2012	Сапоги	10 500
05.11.2012	Молоко	45

База данных

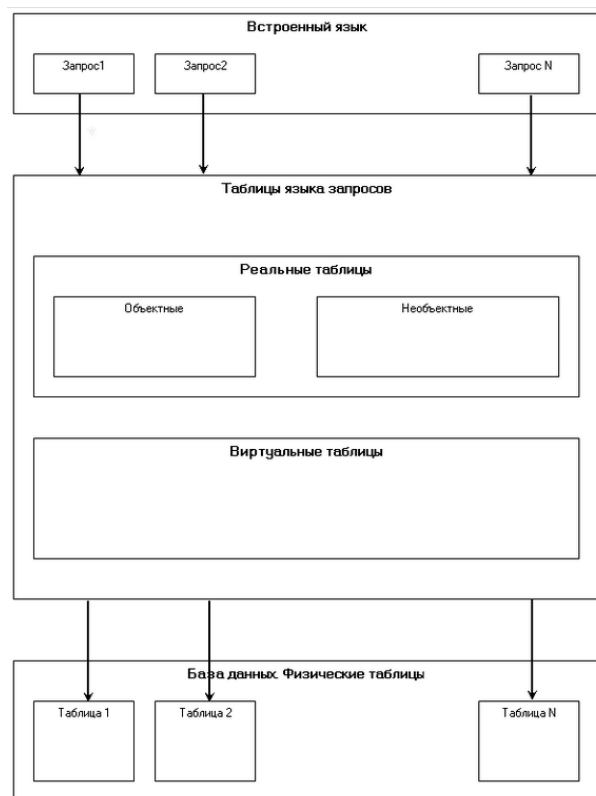
Периодический независимый регистр сведений "Цены"

Период	Товар	Цена
15/10/2012	Туфли	7000
15/10/2012	Сапоги	10000
15/10/2012	Кроссовки	3000
20/10/2012	Масло	70
20/10/2012	Молоко	40
20/10/2012	Сметана	50
01/11/2012	Туфли	8000
01/11/2012	Масло	80
05/11/2012	Сапоги	10500
05/11/2012	Молоко	45

Как видно, ни одна строка регистра не имеет указателя-ссылки на себя. Обращение к записи осуществляется по ключу, в качестве которого выступает совокупность значений измерений. В примере на рисунке это будут поля Период и Товар.

Источники данных для запросов

При работе с технологической платформой нет возможности непосредственного обращения к физическим таблицам базы данных. Объяснение этому кроется в том факте, что при использовании конфигурации могут использоваться в разных случаях различные СУБД, каждая из которых поддерживает свой диалект SQL и свой ANSI стандарт, а текст запроса, интегрируемый в код конфигурации, должен одинаково работать на любой используемой СУБД. Поэтому запрос пишется на встроенном в платформу языке запросов 1С, а при выполнении запроса платформа автоматически транслирует его текст в набор инструкций конкретной СУБД. Кроме того, физические таблицы и поля в них именованы так, что непонятно, что именно хранится в данном поле или таблице. Процесс обращения к физическим данным представлен на рисунке .



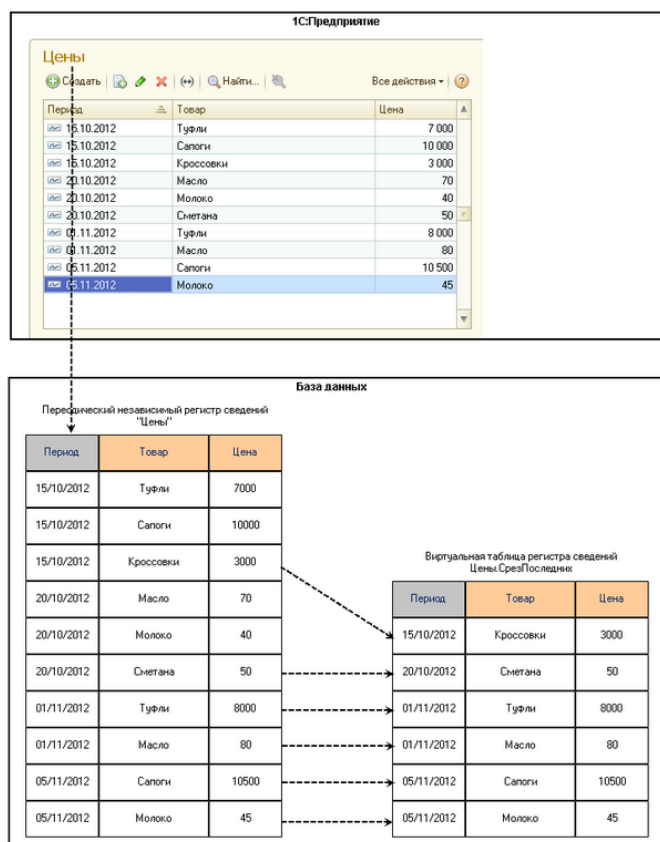
Таким образом, можно сделать вывод, что все таблицы языка запросов существуют временно в адресном пространстве технологической платформы в отличии от **физических** таблиц базы данных, которые существуют на жестком носителе постоянно. По «степени подобия» физическим таблицам они подразделяются на **реальные** и **виртуальные**.

Каждая реальная таблица соответствует конкретной физической таблице. Так, реальная таблица Справочник.Клиенты соответствует справочнику Клиенты, а таблица РегистрСведений.Цены, соответствует регистру сведений Цены. На рисунке ниже представлено сравнение реальной и физической таблиц.

Реальная таблица	Физическая таблица
Справочник.Клиенты	_Reference<n>
Ссылка	_Id<suff>
ВерсияДанных	_Version
ПометкаУдаления	_Marked
Предопределенный	_IsMetadata
Родитель	_ParentId<suff>
Владелец	_OwnerId<suff>
ЭтоГруппа	_Folder
Код	_Code
Наименование	_Description
<Имя реквизита>	_Fld<n><suff>
<Имя общего реквизита>	_Fld<n><suff>
Представление	–
<Имя табличной части>	–

Реальные таблицы подразделяются на объектные (ссылочные) и неobjектные (нессылочные). Отличительной особенностью объектных (ссылочных) таблиц является то, что они включают в себя стандартное поле Ссылка, которое позволяет однозначно идентифицировать каждую запись (данные об объекте базы данных). Эти таблицы могут быть иерархическими, подчиненными, и поля таких таблиц могут содержать вложенные таблицы (табличные части).

Виртуальные таблицы не похожи на какую-либо физическую таблицу и состоят из других записей, нежели реальные таблицы, т.к. формируются в момент выполнения запроса на основе данных реальных таблиц. Здесь в качестве примера можно привести виртуальные таблицы регистров сведений *СрезПоследних*, которые содержат информацию о самых последних записях на конкретный период времени по совокупности значений измерений.



Т.е. при обращении с запросом к виртуальной таблице *СрезПоследних* будут предоставлены не все записи физической таблицы регистра сведений, а лишь последние по времени.

Язык запросов 1С

Запросы предназначены для выборки информации из базы данных. Иными словами, запрос — это «просьба» к системе выбрать информацию из базы данных, и, возможно, сгруппировать, отсортировать, вычислить. Язык запросов 1С:Предприятия 8 - это диалект SQL, содержащий различные расширения, ориентированные на специфику финансово-экономических задач. Посредством запроса к информационной базе можно получить разнообразные выборки данных из одной или нескольких взаимосвязанных таблиц, отобранных и отсортированных определенным образом .

Следует иметь в виду, что с помощью запросов можно исключительно прочитать нужную информацию из базы данных, но изменить, удалить ее или записать обратно при помощи запроса нельзя – для этого используется объектное обращение.

Для работы с запросами предусмотрено несколько специальных объектов:

- **Запрос**
- **РезультатЗапроса**
- **ВыборкаИзРезультатаЗапроса** или **ТаблицаЗначений**

Общая схема выполнения запроса состоит из следующих этапов:

- Создание *объекта Запрос* с нужным текстом запроса на специальном языке запросов
- Установка параметров запроса с помощью метода *УстановитьПараметр*
- Выполнение запроса и получение *объекта РезультатЗапроса* с помощью метода **Выполнить()**
- Получение *объекта Выборка* из объекта *РезультатаЗапроса* с помощью метода **Выбрать()** или выгрузка *РезультатаЗапроса* во *временный объект ТаблицаЗначений* с помощью метода **Выгрузить()**.
- Обработка *Выборки* или *ТаблицыЗначений* - чаще всего это перебор строк или выполнение иных действий, для которых был нужен запрос



Рассмотрим каждый этап схемы:

1. **Установка текста запроса**, который выбирает информацию из справочника Номенклатура о наименовании и единице хранения всех номенклатурных позиций определенного производителя, значение которого задается в качестве параметра запроса (&Производитель).

```
Запрос = Новый Запрос;
Запрос.Текст =
"| ВЫБРАТЬ Наименование, ЕдиницаХранения КАК ЕдХранения
| ИЗ Справочник.Номенклатура
| ГДЕ Производитель = &Производитель";
```

2. **Установка параметра запроса**

```
Запрос.УстановитьПараметр("Производитель",
Справочники.Производители.НайтиПоНаименованию("Основной производитель"));
```

3. **Выполнение запроса и получение результата запроса**

```
РезультатЗапроса = Запрос.Выполнить();
```

4. **Обработка результата**

а) Получение выборки

```
Выборка = РезультатЗапроса.Выбрать();
```

```
Пока Выборка.Следующий() Цикл
    *****Некие действия*****
    Сообщить(Выборка.Наименование);
КонецЦикла;
```

б) Выгрузка результата запроса в таблицу значений

```
таблЗнч = РезультатЗапроса.Выгрузить();
Для Каждого СтрТЗ Из таблЗнч Цикл
    *****Некие действия*****
    Сообщить(Выборка.Наименование);
КонецЦикла
```

Синтаксис текста запросов

Язык запросов «1С:Предприятия» основан на стандартном SQL. Команда ВЫБРАТЬ по структуре подобна команде SQL SELECT, однако имеет свои особенности.

Итак, назначение команды ВЫБРАТЬ — получение результирующего набора данных, который представляет из себя двумерный массив, сформированный из данных входного набора — так же двумерного множества, полученного из одной или нескольких реальных таблиц. Упрощенная структура команды имеет следующий вид:

```
ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] <список_полей>  
ИЗ <входное_множество>  
[ГДЕ <условие_отбора_строк>]  
[СГРУППИРОВАТЬ ПО <список_полей_группировки>]  
[ИМЕЮЩИЕ <условие_отбора_групп>]  
[УПОРЯДОЧИТЬ ПО <список_полей> [порядок]]  
[ИТОГИ <список_итогов> ПО <список_полей>]  
[ОБЪЕДИНИТЬ ВСЕ]
```

* В синтаксисе в квадратных скобках [] указываются необязательные разделы, в угловых скобках < > - синтаксические конструкции.

Раздел ИЗ

Порядок выполнения разделов команды не соответствует порядку их следования в синтаксисе. Так первым выполняется раздел ИЗ, где указывается входной набор данных, который может состоять как из одной источника данных, так и из различных соединений несколько источников, строки которых связываются по равенству значений некоторых полей — **условиям связи**. В качестве источников запроса могут выступать реальные и виртуальные таблицы, а также вложенные запросы. Поле, по которому производится связь, обычно имеет ссылочный тип. Условие связи источников запроса задается в предложении ИЗ, после ключевого слова ПО. Таблицы в разделе ИЗ могут соединяться через **внутреннее соединение**, **левое внешнее соединение**, **правое внешнее соединение** или **полное внешнее соединение**.

При *внутреннем соединении* таблиц в результат запроса попадут только те записи из таблиц-источников, которые удовлетворяют заданному условию связи

Пример 1.

ВЫБРАТЬ

Товары.Код,
Товары.Наименование,
Товары.Производитель,
Цены.Цена

ИЗ Справочник.Товары КАК Товары ВНУТРЕННЕЕ СОЕДИНЕНИЕ

РегистрСведений.Цены.СрезПоследних КАК Цены ПО Товары.Ссылка = Цены.Товар

Запрос: Справочник.Товары (Записей в результате: 6)

Код	Наименование	Производитель	Цена
000000002	Туфли	Германия	8 000
000000003	Сапоги	Италия	10 500
000000004	Кроссовки		3 000
000000006	Масло	Россия	80
000000007	Молоко	Россия	45
000000008	Сметана		50

При *левом соединении*, в результат запроса попадут записи из обеих таблиц, удовлетворяющие условию связи, и, кроме того, записи из первой таблицы, расположенной слева от ключевого слова СОЕДИНЕНИЕ, для которых не найдено соответствия во второй таблице.

Пример 2.

ВЫБРАТЬ

Товары.Код,
Товары.Наименование,
Товары.Производитель,
Цены.Цена

ИЗ Справочник.Товары КАК Товары ЛЕВОЕ ВНЕШНЕЕ

СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних КАК Цены ПО
Товары.Ссылка = Цены.Товар

Запрос: Справочник.Товары (Записей в результате: 10)

Код	Наименование	Производитель	Цена
000000001	Обувь		
000000002	Туфли	Германия	8 000
000000003	Сапоги	Италия	10 500
000000004	Кроссовки		3 000
000000005	Продукты		
000000006	Масло	Россия	80
000000007	Молоко	Россия	45
000000008	Сметана		50
000000009	Детская обувь		
000000010	Пинетки		

При *правом соединении* таблиц ситуация зеркально противоположная. То есть из правой таблицы в результат запроса попадут все записи, а из левой – только те, которые удовлетворяют условию связи.

Пример 3.

ВЫБРАТЬ

Товары.Код,
Товары.Наименование,
Товары.Производитель,
Цены.Цена

ИЗ Справочник.Товары КАК Товары ПРАВОЕ ВНЕШНЕЕ СОЕДИНЕНИЕ

РегистрСведений.Цены.СрезПоследних КАК Цены ПО Товары.Ссылка = Цены.Товар

Запрос: Справочник.Товары (Записей в результате: 6)

Код	Наименование	Производитель	Цена
000000002	Туфли	Германия	8 000
000000003	Сапоги	Италия	10 500
000000004	Кроссовки		3 000
000000006	Масло	Россия	80
000000007	Молоко	Россия	45
000000008	Сметана		50

При *полном соединении* таблиц в результат запроса будут включены записи из обеих исходных таблиц, которые соответствуют указанному условию. Кроме того, в результат запроса будут включены также еще и те записи из обоих источников, для которых не найдено соответствий.

Пример 4.

ВЫБРАТЬ

Товары.Код,

Товары.Наименование,

Товары.Производитель,

Цены.Цена

ИЗ Справочник.Товары КАК Товары ПОЛНОЕ ВНЕШНЕЕ

СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних КАК Цены ПО

Товары.Ссылка = Цены.Товар

Запрос: Справочник.Товары (Записей в результате: 10)

Код	Наименование	Производитель	Цена
000000001	Обувь		
000000002	Туфли	Германия	8 000
000000003	Сапоги	Италия	10 500
000000004	Кроссовки		3 000
000000005	Продукты		
000000006	Масло	Россия	80
000000007	Молоко	Россия	45
000000008	Сметана		50
000000009	Детская обувь		
000000010	Пинетки		

Раздел ВЫБРАТЬ

Раздел ВЫБРАТЬ является обязательным для команды. В нем указывается список полей из исходного набора, которые попадут в результирующий набор. Для удобства обращения к полям им могут быть заданы псевдонимы после ключевого слова КАК. Для исключения одинаковых строк из результирующего набора используется ключевое слово РАЗЛИЧНЫЕ. При использовании ключевого слова РАЗРЕШЕННЫЕ в результирующий набор будут выведены записи, на которые у текущего пользователя есть права.

Пример 4. В примере сравниваются два запроса и относящиеся к ним результирующие наборы данных: один с использованием ключевого слова РАЗЛИЧНЫЕ, а другой без.

ВЫБРАТЬ

ЗаказТовара.Клиент КАК Клиент
ИЗ Документ.ЗаказТовара КАК ЗаказТовара

Номер	Клиент
000000001	Соколов Иван Андреевич
000000002	Соколов Иван Андреевич
000000003	Орлов Сергей Иванович
000000004	Маслова Ирина Николаевна
000000005	Маслова Ирина Николаевна
000000006	Соколов Иван Андреевич
000000007	Орлов Сергей Иванович

ВЫБРАТЬ РАЗЛИЧНЫЕ

ЗаказТовара.Клиент КАК Клиент
ИЗ Документ.ЗаказТовара КАК ЗаказТовара

Клиент
Соколов Иван Андреевич
Орлов Сергей Иванович
Маслова Ирина Николаевна

Раздел ГДЕ

В данном разделе формулируется условие отбора в результирующий набор отдельных строк из исходного множества.

Пример 5.

```

ВЫБРАТЬ
    Накладная.Дата КАК Дата,
    Накладная.Номер,
    Накладная.Поставщик
ИЗ Документ.ПриходнаяНакладная КАК Накладная
ГДЕ Дата >= ДАТАВРЕМЯ(2012, 10, 01) И Дата < ДАТАВРЕМЯ(2012, 11, 01)

```

В условии могут использоваться различные операторы. Например, логические: И, ИЛИ, НЕ, МЕЖДУ, ПОДОБНО; сравнения : >, <, >=, <=, <> и т.д.

*** Узнать обо всех операторах можно в конструкторе запросов, а так же в синтаксис-помощнике, встроенном в платформу.**

Раздел СГРУППИРОВАТЬ ПО

Иногда необходимо получить такие значения, как СУММА, МАКСИМУМ, МИНИМУМ, СРЕДНЕЕ, каждое из которых является обобщающей характеристикой некоторого множества значений. Такие характеристики получаются с помощью соответствующих агрегатных функций: СУММА(), МАКСИМУМ(), МИНИМУМ(), СРЕДНЕЕ(). С помощью раздела СГРУППИРОВАТЬ ПО можно указать условия формирования групп, которые будут выступать аргументами для агрегатных функций.

Пример 6. Вычислить минимальное, максимальное, среднее значение цен для каждого товара, представленного в табличной части *Состав* документа *РасходнаяНакладная*

ВЫБРАТЬ

Накладная.Состав.Товар КАК Товар,
МИНИМУМ(Накладная.Состав.Цена) КАК Минимум,
МАКСИМУМ(Накладная.Состав.Цена) КАК Максимум,
СРЕДНЕЕ(Накладная.Состав.Цена) КАК Среднее

ИЗ Документ.РасходнаяНакладная.Состав КАК Накладная.Состав
СГРУППИРОВАТЬ ПО Накладная.Состав.Товар

Товар	Количество	Цена	Сумма
Туфли	3	6 000	18 000
Туфли	3	8 000	24 000
Туфли	3	7 000	21 000
Туфли	5	6 000	30 000
Сапоги	3	10 000	30 000
Сапоги	1	12 000	12 000
Сапоги	3	11 000	33 000
Сапоги	3	11 000	33 000
Кроссовки	10	3 500	35 000
Кроссовки	5	4 000	20 000
Кроссовки	5	3 000	15 000
Масло	20	100	2 000
Масло	20	90	1 800
Масло	10	90	900
Молоко	20	50	1 000
Молоко	10	60	600
Молоко	10	60	600

Товар	Минимум	Максимум	Среднее
Туфли	6 000	8 000	6 750
Сапоги	10 000	12 000	11 000
Кроссовки	3 000	4 000	3 500
Масло	90	100	93,333333
Молоко	50	60	56,666667

В данном примере множество строк табличной части *Состав* будет разбито на группы по каждому товару и в каждой такой группе будут вычислены минимальное, максимальное, среднее значение.

Раздел ИМЕЮЩИЕ

В разделе ИМЕЮЩИЕ формулируется условия отбора групп в результирующий набор. По своей сути данный раздел аналогичен разделу ГДЕ за тем исключением, что раздел ГДЕ используется для отбора строк и в нем не присутствуют агрегатные функции.

Пример 7. Написать запрос выводящий сведения о покупателе, купившем товаров на сумму более 50000 , а так же количестве и сумме проданного ему товара.

ВЫБРАТЬ

Накладная.Состав.Ссылка.Покупатель КАК Покупатель,

Накладная.Состав.Товар КАК Товар,

СУММА(Накладная.Состав.Количество) КАК Количество,

СУММА(Накладная.Состав.Сумма) КАК Сумма

ИЗ

Документ.Расходная.Накладная.Состав КАК Накладная.Состав

СГРУППИРОВАТЬ ПО

Накладная.Состав.Ссылка.Покупатель,

Накладная.Состав.Товар

ИМЕЮЩИЕ СУММА(Накладная.Состав.Сумма) > 50000

Покупатель	Товар	Количество	Сумма
Соколов Иван Андреевич	Туфли	5	30 000
Соколов Иван Андреевич	Сапоги	3	33 000
Соколов Иван Андреевич	Кроссовки	10	35 000
Орлов Сергей Иванович	Масло	40	3 800
Маслова Ирина Николаевна	Туфли	9	63 000
Орлов Сергей Иванович	Молоко	20	1 000
Маслова Ирина Николаевна	Сапоги	7	75 000
Маслова Ирина Николаевна	Кроссовки	10	35 000
Маслова Ирина Николаевна	Масло	10	900
Маслова Ирина Николаевна	Молоко	20	1 200

Покупатель	Товар	Количество	Сумма
Маслова Ирина Николаевна	Туфли	9	63 000
Маслова Ирина Николаевна	Сапоги	7	75 000

Раздел УПОРЯДОЧИТЬ ПО

Данный раздел позволяет расположить записи в нужном порядке. Упорядочивание результата запроса позволяет предугадать последовательность следования записей в результате, что может дать большой выигрыш как при программировании многих алгоритмов, так и при выполнении запросов. По умолчанию упорядочивание производится по возрастанию ВОЗР, но можно указать и убывание УБЫВ.

Пример 8. Упорядочить записи регистра Цены по убыванию по дате и по возрастанию цены.

ВЫБРАТЬ

Цены.Период КАК Период,

Цены.Товар КАК Товар,

Цены.Цена КАК Цена

ИЗ РегистрСведений.Цены КАК Цены

УПОРЯДОЧИТЬ ПО Период УБЫВ, Цена

Период	Товар	Цена
05.11.2012 0:00:00	Молоко	45
05.11.2012 0:00:00	Сапоги	10 500
01.11.2012 0:00:00	Масло	80
01.11.2012 0:00:00	Туфли	8 000
20.10.2012 0:00:00	Молоко	40
20.10.2012 0:00:00	Сметана	50
20.10.2012 0:00:00	Масло	70
15.10.2012 0:00:00	Кроссовки	3 000
15.10.2012 0:00:00	Туфли	7 000
15.10.2012 0:00:00	Сапоги	10 000

Раздел ИТОГИ ПО

При использовании группировок и агрегатных функций записи в результирующем наборе сначала объединяются в группы, а потом как бы «сворачиваются» по агрегатным характеристикам. При использовании раздела ИТОГИ ПО записи, содержащие одинаковые значения полей, по которым нужно рассчитать итоги, собираются вместе, для каждой группы записей вычисляются значения агрегатных функций. При этом результат вычислений помещается в итоговые строки и добавляется к записям исходной таблицы (детальным записям). То есть детальные записи не сворачиваются, а, наоборот, достраиваются относительно исходной таблицы и служат для детализации итоговых строк.

Общий состав приходных накладных

Запрос: Документ.ПриходнаяНакладная.Состав (Записей в результате: 12)

Товар	Количество	Цена	Сумма
Туфли	5	5 000	25 000
Туфли	10	8 000	80 000
Сапоги	5	10 000	50 000
Сапоги	5	10 000	50 000
Кроссовки	15	3 000	45 000
Кроссовки	20	3 000	60 000
Масло	20	80	1 600
Масло	30	70	2 100
Молоко	30	40	1 200
Молоко	50	40	2 000
Сметана	50	50	2 500
Сметана	20	50	1 000

Результат построения итогов по полю "Товар"

Запрос: Документ.ПриходнаяНакладная.Состав (Записей в результате: 18)

Товар	Количество	Цена	Сумма
Кроссовки	35	3 000	105 000
Кроссовки	15	3 000	45 000
Кроссовки	20	3 000	60 000
Сапоги	10	10 000	100 000
Сапоги	5	10 000	50 000
Сапоги	5	10 000	50 000
Туфли	15	6 500	105 000
Туфли	5	5 000	25 000
Туфли	10	8 000	80 000
Масло	50	75	3 700
Масло	20	80	1 600
Масло	30	70	2 100
Молоко	80	40	3 200
Молоко	30	40	1 200
Молоко	50	40	2 000
Сметана	70	50	3 500
Сметана	50	50	2 500
Сметана	20	50	1 000

Результат группировки по полю "Товар"

Запрос: Документ.ПриходнаяНакладная.Состав (Записей в результате: 6)

Товар	Количество	Цена	Сумма
Туфли	15	6 500	105 000
Сапоги	10	10 000	100 000
Кроссовки	35	3 000	105 000
Масло	50	75	3 700
Молоко	80	40	3 200
Сметана	70	50	3 500

Пример 9.

ВЫБРАТЬ

НакладнаяСостав.Товар КАК Товар,
 НакладнаяСостав.Количество КАК Количество,
 НакладнаяСостав.Цена КАК Цена,
 НакладнаяСостав.Сумма КАК Сумма

ИЗ

ИТОГИ

СУММА(Количество),
 СРЕДНЕЕ(Цена),
 СУММА(Сумма)

ПО Товар

Товар	Количество	Цена	Сумма
Кроссовки	35	3 000	105 000
Кроссовки	15	3 000	45 000
Кроссовки	20	3 000	60 000
Сапоги	10	10 000	100 000
Сапоги	5	10 000	50 000
Сапоги	5	10 000	50 000
Туфли	15	6 500	105 000
Туфли	5	5 000	25 000
Туфли	10	8 000	80 000
Масло	50	75	3 700
Масло	20	80	1 600
Масло	30	70	2 100
Молоко	80	40	3 200
Молоко	30	40	1 200
Молоко	50	40	2 000
Сметана	70	50	3 500
Сметана	50	50	2 500
Сметана	20	50	1 000

Раздел ОБЪЕДИНИТЬ ВСЕ

Раздел позволяет объединить результирующие наборы нескольких запросов в один. При объединении каждый запрос получает данные независимо, то есть у каждого из запросов – свое описание выбираемых полей (ВЫБРАТЬ), источников запроса (ИЗ), условий отбора (ГДЕ), полей группировки (СГРУППИРОВАТЬ ПО). Затем данные, получаемые в результате каждого запроса, объединяются, и уже над этим объединением выполняются такие операции, как упорядочивание результатов (УПОРЯДОЧИТЬ ПО) и расчет итогов (ИТОГИ ПО). Объединяемые запросы должны иметь одинаковое количество полей в списке полей выборки. В случае, если поля выборки объединяемых запросов имеют разный тип, поля результата запроса будут иметь составной тип.

Пример 10. Написать запрос, объединяющий вместе информацию из документов Заказ товара и информацию из документов Расходная накладная. При этом нужно вывести суммарное количество заказанных и проданных товаров по каждому клиенту. Также мы хотим видеть общий итог по результату запроса в целом и отдельно по каждому клиенту.

ВЫБРАТЬ

Заказ.Ссылка.Клиент КАК Клиент,
Заказ.Товар КАК Товар,
СУММА(Заказ.Количество) КАК Заказано,
СУММА(0) КАК Продано

ИЗ Документ.ЗаказТовара.Состав КАК Заказ

СГРУППИРОВАТЬ ПО Заказ.Ссылка.Клиент, Заказ.Товар

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

Накладная.Ссылка.Покупатель,
Накладная.Товар,
СУММА(0),
СУММА(Накладная.Количество)

ИЗ Документ.РасходнаяНакладная.Состав КАК Накладная

СГРУППИРОВАТЬ ПО Накладная.Ссылка.Покупатель,

Накладная.Товар

ИТОГИ ПО Клиент

Клиент	Товар	Заказано	Продано
		191	134
Соколов Иван Андреевич		26	18
Соколов Иван Андреевич	Туфли	6	0
Соколов Иван Андреевич	Сапоги	5	0
Соколов Иван Андреевич	Кроссовки	15	0
Соколов Иван Андреевич	Туфли	0	5
Соколов Иван Андреевич	Сапоги	0	3
Соколов Иван Андреевич	Кроссовки	0	10
Орлов Сергей Иванович		100	60
Орлов Сергей Иванович	Масло	30	0
Орлов Сергей Иванович	Молоко	30	0
Орлов Сергей Иванович	Сметана	40	0
Орлов Сергей Иванович	Масло	0	40
Орлов Сергей Иванович	Молоко	0	20
Маслова Ирина Николаевна		65	56
Маслова Ирина Николаевна	Туфли	10	0
Маслова Ирина Николаевна	Сапоги	10	0
Маслова Ирина Николаевна	Масло	25	0
Маслова Ирина Николаевна	Сметана	20	0
Маслова Ирина Николаевна	Туфли	0	9
Маслова Ирина Николаевна	Сапоги	0	7
Маслова Ирина Николаевна	Кроссовки	0	10
Маслова Ирина Николаевна	Масло	0	10
Маслова Ирина Николаевна	Молоко	0	20