

Дистанционный тренинг

# **Разработка и оптимизация запросов в 1С:Предприятие 8.3**

Версия книги: 1.2

## Оглавление

Введение .....	8
Содержание курса .....	8
Требования для прохождения курса .....	8
Компоненты курса .....	9
Рекомендуемая схема обучения .....	9
Модуль 1. Общая информация о хранении данных в «1С:Предприятие 8» .....	10
Получение данных из базы .....	10
Модели доступа к данным .....	10
Объектная модель доступа к данным .....	13
Преимущества и ограничения объектного доступа .....	15
Табличная модель доступа к данным .....	16
Таблицы объектов метаданных .....	18
Константы .....	18
Планы обмена .....	19
Справочники .....	19
Документы .....	19
Последовательности .....	19
Журналы документов .....	19
Перечисления .....	19
Планы видов характеристик .....	19
Планы счетов .....	20
Планы видов расчета .....	20
Регистры сведений .....	20
Регистры накопления .....	21
Регистры бухгалтерии .....	21
Регистры расчета .....	22
Бизнес-процессы .....	22
Задачи .....	22
Физические и виртуальные таблицы .....	25
Физические таблицы .....	26
Виртуальные таблицы .....	26
Внешние источники данных .....	27
Модуль 2. Синтаксис текста запроса .....	30
Язык запросов .....	30
Сравнение языка запросов с SQL .....	30
Структура запроса .....	35
Видеоурок. Консоль запросов .....	37
Видеоурок. Простой запрос .....	37
Практическое задание №1. Вычисление выражений .....	38
Видеоурок. Конструктор запроса .....	39
Видеоурок. Таблицы и поля .....	39
Псевдонимы полей .....	40
Комментарии в языке запросов .....	41
Группировка результатов запроса .....	41
Практическое задание №2. Использование агрегатных функций .....	44
Практическое задание №3. Определение количества записей в таблице базы данных .....	45
Практическое задание №4. Использование агрегатных функций для определения количества документов, делающих движения в регистр .....	46

Упорядочивание результатов запроса .....	47
Автоупорядочивание .....	49
Видеоурок. Текстовое представление ссылочного поля .....	51
Видеоурок. Функция ПРЕДСТАВЛЕНИЕ() .....	51
Отличие поля Представление от функции ПРЕДСТАВЛЕНИЕ() .....	52
Ключевое слово ПЕРВЫЕ .....	53
Практическое задание №5. Получение определенного количества записей в запросе .....	54
Ключевое слово РАЗЛИЧНЫЕ .....	55
Видеоурок. Условия в тексте запроса .....	57
Практическое задание №6. Условия в запросе .....	58
Оператор ПОДОБНО .....	59
Практическое задание №7. Использование спецсимвола в запросе .....	61
Практическое задание №8. Использование оператора ПОДОБНО .....	62
Видеоурок. Использование параметров .....	63
Практическое задание №9. Использование параметров и ключевого слова РАЗЛИЧНЫЕ в запросе .....	64
Практическое задание №10. Работа со строками .....	65
Практическое задание №11. Работа с подчиненными справочниками .....	66
Видеоурок. Замена таблицы в запросе при помощи конструктора .....	67
Видеоурок. Использование вложенных таблиц в запросе .....	67
Табличные части справочников и документов в запросах .....	68
Функция ЗНАЧЕНИЕ() .....	69
Практическое задание №12. Использование в запросах предопределенных данных конфигурации .....	71
Видеоурок. Функции для работы с датами .....	72
Практическое задание №13. Использование ключевого слова РАЗЛИЧНЫЕ .....	73
Видеоурок. Выбор данных из иерархических справочников .....	74
Видеоурок. Работа с иерархическими справочниками .....	74
Видеоурок. Использование конструкции ИМЕЮЩИЕ .....	75
Практическое задание №14. Применение условий к агрегатным функциям .....	76
Видеоурок. Оператор ВЫБОР .....	77
Практическое задание №15. Использование оператора ВЫБОР .....	78
Видеоурок. Строки неограниченной длины .....	79
Типы данных в запросах .....	79
Видеоурок. Особенность предложения ИЗ .....	83
Видеоурок. Особенность СГРУППИРОВАТЬ ПО .....	83
Ключевое слово РАЗРЕШЕННЫЕ .....	83
Видеоурок. Особенности конструктора запросов .....	85
Видеоурок. Управляемый конструктор запроса .....	85
Видеоурок. Объектная модель запроса .....	86
Видеоурок. Отображение плана запроса в консоли запросов .....	86
Практическое задание №16. Разыменованное поле .....	88
Практическое задание №17. Получение запросом даты документа-регистратора из регистра .....	89
Практическое задание №18. Переопределение основного представления справочника в платформе 8.3. Виртуальное поле Представление в запросе .....	90
Модуль 3. Работа с итогами в тексте запроса .....	92
Предназначение итогов .....	92
Видеоурок. Предназначение итогов в запросе. Отличие итогов от группировки .....	95
Предложение ИТОГИ .....	96
Практическое задание №19. Применение итогов в запросах .....	97
Итоги для иерархических справочников .....	98
Получение итогов по нескольким полям .....	100

Видеоурок. Совместное использование итогов и группировки	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Общие итоги.....	<b>Ошибка! Закладка не определена.</b>
Модуль 4. Использование нескольких таблиц в запросе: вложенные запросы, соединение, объединение .....	<b>Ошибка! Закладка не определена.</b>
Назначение вложенных запросов .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Редактирование вложенных запросов при помощи конструктора... <b>не определена.</b>	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Оператор В – проверка совпадения значений.....	<b>Ошибка! Закладка не определена.</b>
Практическое задание №20. Использование оператора В .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Упорядочивание во вложенных запросах .....	<b>Ошибка! Закладка не определена.</b>
Итоги во вложенных запросах .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Соединение в запросах.....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Декартово произведение .....	<b>Ошибка! Закладка не определена.</b>
Виды соединения таблиц.....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Внутреннее соединение.....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Левое (внешнее) соединение .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Правое (внешнее) соединение.....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Полное (внешнее) соединение.....	<b>Ошибка! Закладка не определена.</b>
Производительность полного соединения при использовании СУБД PostgreSQL.....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Значение NULL .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. NULL для реквизитов битой ссылки.....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Отладка запросов при помощи выгрузки в таблицу значений ....	<b>Ошибка! Закладка не определена.</b>
Разыменованное полей в запросе .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Неоднозначные поля в запросе .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Особенность левого соединения .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Автоматическая подстановка соединения в конструкторе запроса	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Как соединить две таблицы, если у них нет общего поля .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Объединение в запросах .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Объединение запросов в конструкторе запроса ..	<b>Ошибка! Закладка не определена.</b>
Видеоурок. ОБЪЕДИНИТЬ ВСЕ и ОБЪЕДИНИТЬ .....	<b>Ошибка! Закладка не определена.</b>
Одновременное использование ОБЪЕДИНИТЬ ВСЕ и ОБЪЕДИНИТЬ в запросе .....	<b>Ошибка! Закладка не определена.</b>
Ключевое слово ПУСТАЯТАБЛИЦА .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Полезный прием объединения таблиц.....	<b>Ошибка! Закладка не определена.</b>
Отличие соединения таблиц от объединения .....	<b>Ошибка! Закладка не определена.</b>
Модуль 5. Работа с запросами при помощи встроенного языка.	<b>Ошибка! Закладка не определена.</b>
Типы данных для работы с запросами .....	<b>Ошибка! Закладка не определена.</b>
Передача параметров в запрос .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Результат выполнения запроса и выборка .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Обход результата запроса .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Функции Следующий() и СледующийПоЗначениюПоля().....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Обход результата запроса по группировкам .....	<b>Ошибка! Закладка не определена.</b>
Видеоурок. Иерархический обход результата запроса .....	<b>Ошибка! Закладка не определена.</b>
Схематическое отображение вариантов обхода результата запроса .....	<b>Ошибка! Закладка не определена.</b>

- Видеоурок. Обход результата запроса, содержащего записи табличной части **Ошибка! Закладка не определена.**
- Видеоурок. Использование конструктора запроса с обработкой результата **Ошибка! Закладка не определена.**
- Видеоурок. Вывод результата запроса в табличный документ ..... **Ошибка! Закладка не определена.**
- Видеоурок. Вывод результата запроса в диаграмму ..... **Ошибка! Закладка не определена.**
- Видеоурок. Анимированные диаграммы в платформе 8.3.5... **Ошибка! Закладка не определена.**
- Видеоурок. Построение кросс-таблицы ..... **Ошибка! Закладка не определена.**
- Видеоурок. Пример построения кросс-таблицы ..... **Ошибка! Закладка не определена.**
- Видеоурок. Обход результата запроса с дополнительной детализацией по периоду ..... **Ошибка! Закладка не определена.**
- Видеоурок. Дополнительные периоды в итогах по дате (пример) ..... **Ошибка! Закладка не определена.**
- Видеоурок. Выгрузка результата запроса в коллекции ..... **Ошибка! Закладка не определена.**
- Видеоурок. Отладка запросов при помощи выгрузки в таблицу значений .... **Ошибка! Закладка не определена.**
- Модуль 6. Использование временных таблиц и пакетных запросов ..... **Ошибка! Закладка не определена.**
- Назначение временных таблиц ..... **Ошибка! Закладка не определена.**
- Менеджер временных таблиц ..... **Ошибка! Закладка не определена.**
- Видеоурок. Использование временных таблиц ..... **Ошибка! Закладка не определена.**
- Видеоурок. Уничтожение временных таблиц ..... **Ошибка! Закладка не определена.**
- Видеоурок. Запрос к временной таблице ..... **Ошибка! Закладка не определена.**
- Видеоурок. Индексирование временной таблицы ..... **Ошибка! Закладка не определена.**
- Видеоурок. Передача внешней таблицы в запрос ..... **Ошибка! Закладка не определена.**
- Видеоурок. Ограничения при передаче таблицы значений в запрос ..... **Ошибка! Закладка не определена.**
- Видеоурок. Работа с временными таблицами в отладчике .... **Ошибка! Закладка не определена.**
- Видеоурок. Использование таблиц значений и параметров в универсальных запросах... **Ошибка! Закладка не определена.**
- Назначение пакета запросов ..... **Ошибка! Закладка не определена.**
- Создание пакета запросов при помощи конструктора ..... **Ошибка! Закладка не определена.**
- Практическое задание №21. Работа с периодическим регистром сведений **Ошибка! Закладка не определена.**
- Выполнение запросов пакета ..... **Ошибка! Закладка не определена.**
- Модуль 7. Использование виртуальных таблиц в запросе ..... **Ошибка! Закладка не определена.**
- Реальные и виртуальные таблицы ..... **Ошибка! Закладка не определена.**
- Структура регистра сведений ..... **Ошибка! Закладка не определена.**
- Видеоурок. Получение срезов для регистров ..... **Ошибка! Закладка не определена.**
- Видеоурок. Итоги регистров сведений на платформе 8.3 ..... **Ошибка! Закладка не определена.**
- Видеоурок. Параметры у виртуальных таблиц среза ..... **Ошибка! Закладка не определена.**
- Практическое задание №22. Особенности получения среза последних ..... **Ошибка! Закладка не определена.**
- Практическое задание №23. Особенности использования параметров виртуальной таблицы среза последних ..... **Ошибка! Закладка не определена.**
- Практическое задание №24. Объединение запросов ..... **Ошибка! Закладка не определена.**
- Структура регистра накопления ..... **Ошибка! Закладка не определена.**
- Получение остатков ..... **Ошибка! Закладка не определена.**
- Видеоурок. Параметры виртуальной таблицы остатков ..... **Ошибка! Закладка не определена.**
- Устройство виртуальной таблицы остатков ..... **Ошибка! Закладка не определена.**

- Видеоурок. Получение оборотов ..... **Ошибка! Залкадка не определена.**
- Практическое задание №25. Вывод в табличный документ результата запроса с несколькими уровнями итогов ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Параметры виртуальной таблицы оборотов ..... **Ошибка! Залкадка не определена.**
- Практическое задание №26. Построение кросс-таблицы без использования СКД ..... **Ошибка! Залкадка не определена.**
- Практическое задание №27. Использование объединения запросов..... **Ошибка! Залкадка не определена.**
- Практическое задание №28. Получение процентов от общего количества запросом ..... **Ошибка! Залкадка не определена.**
- Устройство виртуальной таблицы Обороты для остаточных регистров..... **Ошибка! Залкадка не определена.**
- Устройство виртуальной таблицы Обороты для оборотных регистров ..... **Ошибка! Залкадка не определена.**
- Практическое задание №29. Нужно ли использовать группировку в запросе к виртуальной таблице?..... **Ошибка! Залкадка не определена.**
- Получение остатков и оборотов ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Параметры виртуальной таблицы ОстаткиИОбороты..... **Ошибка! Залкадка не определена.**
- Устройство виртуальной таблицы ОстаткиИОбороты ..... **Ошибка! Залкадка не определена.**
- Структура регистра бухгалтерии..... **Ошибка! Залкадка не определена.**
- Таблица значений субконто..... **Ошибка! Залкадка не определена.**
- Видеоурок. Виртуальная таблица ДвиженияССубконто..... **Ошибка! Залкадка не определена.**
- Видеоурок. Получение остатков ..... **Ошибка! Залкадка не определена.**
- Параметры виртуальной таблицы остатков ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Предназначение параметра Субконто виртуальной таблицы .... **Ошибка! Залкадка не определена.**
- Видеоурок. Использование параметра Период для виртуальной таблицы остатков..... **Ошибка! Залкадка не определена.**
- Получение оборотов из регистра бухгалтерии ..... **Ошибка! Залкадка не определена.**
- Параметры виртуальной таблицы оборотов ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Особенности использования дат при получении оборотов..... **Ошибка! Залкадка не определена.**
- Видеоурок. Получение оборотов между корреспондирующими счетами ..... **Ошибка! Залкадка не определена.**
- Получение остатков и оборотов ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Периодичность таблицы остатков и оборотов..... **Ошибка! Залкадка не определена.**
- Видеоурок. Развернутые остатки..... **Ошибка! Залкадка не определена.**
- Регистры расчетов. Таблицы, доступные для регистров расчета ..... **Ошибка! Залкадка не определена.**
- Основная таблица регистра расчета ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Получение данных графика ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Получение данных базы ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Параметры виртуальной таблицы получения базы ..... **Ошибка! Залкадка не определена.**
- Видеоурок. Получение фактического периода действий ..... **Ошибка! Залкадка не определена.**
- Перерасчеты ..... **Ошибка! Залкадка не определена.**
- Модуль 8. Оптимизация запросов..... **Ошибка! Залкадка не определена.**
- Актуальность задачи оптимизации запросов..... **Ошибка! Залкадка не определена.**
- Включать в выходные поля только необходимые данные ..... **Ошибка! Залкадка не определена.**
- Выбирать объектную или табличную методику получения данных ..... **Ошибка! Залкадка не определена.**

Видеоурок. Конструкция ДЛЯ ИЗМЕНЕНИЯ .....	<b>Ошибка! Залкадка не определена.</b>
Индексирование таблиц.....	<b>Ошибка! Залкадка не определена.</b>
Видеоурок. Соответствие условий запросов и существующих индексов в таблицах БД... <b>Ошибка! Залкадка не определена.</b>	
Соединение с подзапросами .....	<b>Ошибка! Залкадка не определена.</b>
Соединение с виртуальными таблицами .....	<b>Ошибка! Залкадка не определена.</b>
Использование ИЛИ в условиях.....	<b>Ошибка! Залкадка не определена.</b>
Получение данных через точку от полей составного типа .....	<b>Ошибка! Залкадка не определена.</b>
Использование параметров виртуальных таблиц.....	<b>Ошибка! Залкадка не определена.</b>
Не использовать запрос в цикле .....	<b>Ошибка! Залкадка не определена.</b>
Практическое задание №30. Выбор оптимального варианта получения наименования справочника запросом .....	<b>Ошибка! Залкадка не определена.</b>
Не использовать функции от параметров запроса .....	<b>Ошибка! Залкадка не определена.</b>
Не использовать получение поля Ссылка через точку от ссылочных данных... <b>Ошибка! Залкадка не определена.</b>	
Не применять избыточное агрегирование при использовании виртуальных таблиц .....	<b>Ошибка! Залкадка не определена.</b>
Не использовать вывод ссылочных полей в отчет .....	<b>Ошибка! Залкадка не определена.</b>
Избегать запросов к пустым таблицам в режиме автоматических блокировок . <b>Ошибка! Залкадка не определена.</b>	
Запрос изменяется при использовании RLS (ограничение прав на уровне записей).....	<b>Ошибка! Залкадка не определена.</b>
Модуль 9.1. Сложные случаи написания запросов .....	<b>Ошибка! Залкадка не определена.</b>
Срез последних на каждую дату .....	<b>Ошибка! Залкадка не определена.</b>
Срез последних при помощи запроса .....	<b>Ошибка! Залкадка не определена.</b>
Генерация последовательности чисел и дат .....	<b>Ошибка! Залкадка не определена.</b>
Практическое задание №31. Генерация последовательности дат запросом <b>Ошибка! Залкадка не определена.</b>	
Практическое задание №32. Получение списка високосных годов запросом .... <b>Ошибка! Залкадка не определена.</b>	
Нумерация строк в запросе .....	<b>Ошибка! Залкадка не определена.</b>
Остатки в периоде, когда не было движений.....	<b>Ошибка! Залкадка не определена.</b>
Количество дней наличия товара на складе.....	<b>Ошибка! Залкадка не определена.</b>
Выбор значений из регистра сведений за период (описание задачи).....	<b>Ошибка! Залкадка не определена.</b>
Расчет накопительных скидок .....	<b>Ошибка! Залкадка не определена.</b>
Получить первую попавшуюся запись .....	<b>Ошибка! Залкадка не определена.</b>
Аналог оператора EXISTS из SQL .....	<b>Ошибка! Залкадка не определена.</b>
Последнее поступление по каждому товару.....	<b>Ошибка! Залкадка не определена.</b>
Модуль 9.2. Кейсы по запросам .....	<b>Ошибка! Залкадка не определена.</b>
Динамика курсов валют .....	<b>Ошибка! Залкадка не определена.</b>
Динамика курсов валют с количеством дней одинакового курса.....	<b>Ошибка! Залкадка не определена.</b>
Расчет премии сотрудникам.....	<b>Ошибка! Залкадка не определена.</b>
Взаиморасчеты по направлениям деятельности .....	<b>Ошибка! Залкадка не определена.</b>
Списание по партиям .....	<b>Ошибка! Залкадка не определена.</b>
Список потенциальных покупателей .....	<b>Ошибка! Залкадка не определена.</b>
Перечень давно не обращавшихся клиентов .....	<b>Ошибка! Залкадка не определена.</b>
Контроль оперативности обработки заказов .....	<b>Ошибка! Залкадка не определена.</b>
Руководители компаний-должников .....	<b>Ошибка! Залкадка не определена.</b>
Склад карантина .....	<b>Ошибка! Залкадка не определена.</b>

Отчет по продажам..... **Ошибка! Закладка не определена.**  
Отчет по доходам и расходам..... **Ошибка! Закладка не определена.**  
Практическое задание №33. Выполнение различных запросов в зависимости от условий  
..... **Ошибка! Закладка не определена.**  
Практическое задание №34. Выгрузка запроса в дерево значений ..... **Ошибка! Закладка не определена.**  
Практическое задание №35. Получение данных из внешней информационной базы ..... **Ошибка! Закладка не определена.**  
Практическое задание №36. Использование запросов для вывода дополнительных данных в обычных формах ..... **Ошибка! Закладка не определена.**  
Практическое задание №37. Распределение суммы по базе. **Ошибка! Закладка не определена.**  
Практическое задание №38. Создание внешнего источника данных для работы с журналом регистрации. Получение данных из внешнего источника ..... **Ошибка! Закладка не определена.**

## Введение

В данном методическом пособии представлены материалы курса **Разработка и оптимизация запросов в 1С:Предприятие 8.2 и 8.3**.

### Содержание курса

Этот курс посвящен языку запросов в «1С:Предприятие 8». Это очень важная для разработчиков тема, поскольку запросы – один из базовых механизмов платформы, который предназначен для выборки сведений из информационной базы.

Запросы представляют собой обращения к системе для получения определенной информации из базы данных, а также выполнение обработки выбранных данных – например, наложение фильтров, группировка, сортировка.

Запросы являются основой для построения отчетов. При помощи запросов, например, можно получить из информационной базы объем продаж товаров в количественном и суммовом выражении с детализацией по месяцам, отобрать документы, удовлетворяющие определенным условиям и т.д.

На практике запросы могут использоваться, к примеру, в форме подбора номенклатуры, чтобы отобразить остатки товаров на складах и актуальные цены, при проведении документов для контроля остатков и для подбора партии товара, которую необходимо списать.

Запросы используются в большинстве алгоритмов типовых конфигураций. Поэтому очень важно не только уметь самостоятельно писать запросы, но и разбираться в написанных запросах, вносить в них коррективы.

В курсе рассматривается синтаксис языка запросов, отмечаются его особенности, а также сходства и различия с SQL – классическим языком, применяемым для создания, модификации и управления данными в реляционных базах данных. Разбирается работа с запросами при помощи встроенного языка, обработка результатов выполнения запросов. Подробно рассматриваются виртуальные таблицы объектов метаданных, а также даются рекомендации по написанию запросов с точки зрения оптимизации быстродействия системы. Особое место уделяется практике – рассматриваются примеры написания запросов различной сложности.

### Требования для прохождения курса

Курс не накладывает специфичных требований на уровень подготовленности участников обучения – рассматриваемые в нем темы и вопросы окажутся одинаково полезными и для новичков в этой области знаний, и для уже состоявшихся профессионалов.

Тем не менее, в процессе обучения Вам потребуются базовые навыки разработки в «1С:Предприятие 8», нужно быть знакомым с основными объектами метаданных, понимать основные синтаксические конструкции встроенного языка, иметь опыт написания программных модулей, обработчиков событий. Также необходимо иметь базовые представления об устройстве управляемых форм, основных директивах компиляции, клиент-серверном взаимодействии.

Для прохождения курса будет достаточно учебной версии платформы «1С:Предприятие 8». Наличие типовых решений у слушателя желательно, но не обязательно. В курсе рассматриваются примеры работы с запросами в типовых решениях:

- 1С:Управление торговлей, редакция 11
- 1С:Бухгалтерия 8, редакция 3.0
- 1С:Зарплата и управление персоналом 3.0.

## Компоненты курса

- **Данное методическое пособие**  
Поставляется в формате pdf, его можно скачать с сайта.
- **Видеоуроки**  
Некоторые темы курса реализованы в формате видеоуроков. Скачать все видеоуроки тренинга можно с сайта: <http://kursy-po-1c.ru/zapros/materials>
- **Практические задания**  
Для закрепления изученного материала важно выполнять практические задания. После выполнения каждого из них проведите самоконтроль – ознакомьтесь с решением преподавателя.
- **Кейсы и примеры сложных случаев написания запросов**  
Кейс – разбор конкретной практической ситуации с выбором варианта решения и анализом результатов. Кейсы представлены в формате видео, в них реализованы приемы написания запросов, рассмотренные в теоретической части курса.

Скачать данные видеоуроки можно здесь: <http://kursy-po-1c.ru/zapros/materials>

## Рекомендуемая схема обучения

Курс построен от методического пособия. Читайте уроки и главы этой книги.

В процессе чтения Вам будут встречаться уроки в формате видео – переходите к просмотру видео.

После изучения ряда уроков будет предложено выполнить практические задания – скачивайте их со страницы материалов тренинга: <http://kursy-po-1c.ru/zapros/materials>

Желаем успехов в обучении!

# Модуль 1. Общая информация о хранении данных в «1С:Предприятие 8»

## Получение данных из базы

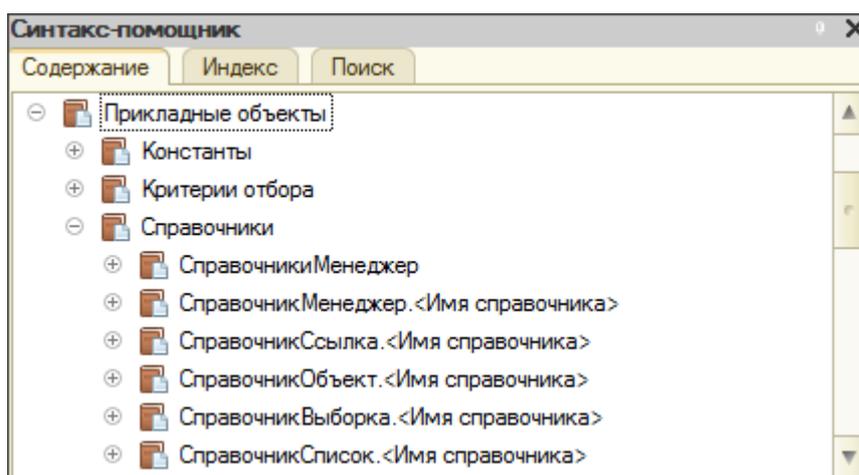
Рассмотрим типичный сценарий работы пользователей с конфигурациями на платформе «1С:Предприятие 8». Вначале пользователи вносят в систему данные при помощи разработанных форм ввода. Посредством встроенного языка выполняется запись введенных данных в базу, где они и хранятся. Это и нормативно-справочная информация, и документы, отражающие факты хозяйственной деятельности предприятия. Так, например, для торговой компании в справочники вносятся клиенты и поставщики, продаваемые товары, виды прайс-листов, собственные юридические лица и т.д. При помощи документов фиксируются закупки товаров от поставщиков, отгрузки товаров клиентам, установки цен и скидок и т.д. Далее на основе введенных данных пользователям требуется получить из базы сводные данные. Это могут быть данные для формирования выходных печатных форм, например, таких как ТОРГ-12 или счет-фактура. Это могут быть различные аналитические отчеты – сведения о рентабельности продаж, валовой прибыли, классификация клиентов по приносимой прибыли, отчет о долгах клиентов и т.д.

Умение корректно и оптимально получать данные из информационной базы, а также представлять их в определенном виде очень важно для разработчика. Большинство автоматизируемых предприятий имеет некоторое количество собственных уникальных печатных форм и управленческих отчетов, отсутствующих в типовых конфигурациях. Эти работы выполняются как на крупных проектах, так и при небольших внедрениях и постоянно востребованы заказчиками. Для решения указанных задач в платформе «1С:Предприятие 8» существуют мощные механизмы – язык запросов и система компоновки данных. Первый из них и является темой текущего курса.

## Модели доступа к данным

К любым данным, хранящимся в информационной базе, можно обратиться двумя способами: либо используя объектную модель доступа к данным, либо используя табличную модель.

При использовании объектной модели доступа к данным сама информационная база представляется как иерархия классов, каждый из которых обладает собственным набором свойств и методов. Например, для справочников в системе присутствуют различные классы.



При работе со справочниками сначала происходит обращение к классу «СправочникиМенеджер», далее получается класс «СправочникМенеджер», от которого можно получить класс «СправочникСсылка», содержащий ссылку на уже конкретный элемент, и т.д. Таким образом, необходимые значения могут быть получены при помощи обращения к тем или иным свойствам и методам.

При использовании табличной модели доступа к данным информационная база представляется как набор таблиц в базе данных.

Рассмотрим, как хранится информация о конкретном товаре в базе данных (элементе справочника «Товары»):

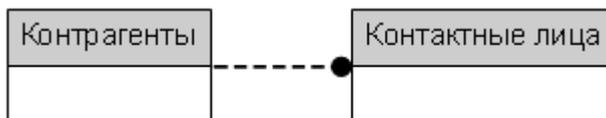
Данные каждого элемента справочника «Товары» – это отдельная строка в таблице базы данных. Ниже описано, какие колонки присутствуют в этой таблице, для хранения каких объектов предназначены, как эти колонки называются, а также какие значения хранятся в этой таблице для указанного выше элемента справочника:

Название колонки в терминах объекта метаданных	Название колонки в базе данных	Значение поля для выбранного элемента справочника
Ссылка	_IDRRef	5198ae43-90eb-11db-a9b0-00055d49b45e
ВерсияДанных	_Version	AAAAAQAAABY=
ПометкаУдаления	_Marked	Ложь
Родитель	_ParentIDRRef	bbb079ae-8c51-11db-a9b0-00055d49b45e
ЭтоГруппа	_Folder	Ложь
Код	_Code	000000006
Наименование	_Description	Ботинки
Артикул	_Fld23	ОБ-003
Поставщик	_Fld109RRef	9d5c4235-8c4c-11db-a9b0-00055d49b45e
ФайлКартинки	_Fld119RRef	df74aa18-3e85-11dc-a9d7-00179a069b10
Вид	Fld120RRef	Товар
Штрихкод	_Fld132	
Описание	_Fld367	
Предопределенный	_IsMetadata	Ложь
ИмяПредопределенных Данных	_PredefinedID	

Тогда строка таблицы базы данных, описывающая выбранный элемент справочника «Товары», будет выглядеть следующим образом:

Ссылка	ВерсияДанных	Это Группа	Код	Наименование
5198ae43-90eb-11db-a9b0-00055d49b45e	AAAAAQAAABY=	Нет	000000006	Ботинки

Таблицы могут быть связаны между собой различными отношениями. Например, в базе данных таблицы *Контрагенты* и *КонтактныеЛица* могут быть связаны между собой связью один-ко-многим. Это означает, что каждое контактное лицо одновременно принадлежит только одному контрагенту, но контрагент может иметь множество контактных лиц.



Нужная информация может быть получена путем обращения к строкам и колонкам конкретных таблиц.

## Объектная модель доступа к данным

При объектной модели доступа к данным работа ведется с помощью объектов встроенного языка, а сама информационная база представляется как набор взаимосвязанных классов.

Рассмотрим несколько примеров объектного доступа к данным.

### Пример 1.

Поиск элемента справочника «Товары по наименованию»

Для поиска используется метод *НайтиПоНаименованию()* объекта типа *СправочникМенеджер.Товары*:

```
НайденныйТовар = Справочники.Товары.НайтиПоНаименованию("Валенки");
```

### Пример 2.

Выборка элементов справочника «Контрагенты», расположенных в определенной папке

Для решения используется метод *Выбрать()* объекта типа *СправочникМенеджер.Контрагенты*. В этом методе первый параметр (*Родитель*) предназначен для фильтрации по родителю, т.е. будут получены элементы в заданной папке. В примере ссылка на папку задается в переменной *Группа*:

```
ВыборкаКонтрагентов = Справочники.Контрагенты.Выбрать(Группа);
```

```
Пока ВыборкаКонтрагентов.Следующий() Цикл
```

```
Сообщение = Новый СообщениеПользователю;
```

```
Сообщение.Текст = ВыборкаКонтрагентов.Наименование;
```

```
Сообщение.Сообщить();
```

```
КонецЦикла;
```

### Пример 3.

**Выборка элементов справочника «Контрагенты», относящихся к указанному региону**

У справочника «Контрагенты» есть реквизит *Регион* (тип – *СправочникСсылка.Регионы*). Чтобы отобразить только тех контрагентов, у которых этот реквизит заполнен нужным значением, используется метод *Выбрать()* объекта типа *СправочникМенеджер.Контрагенты*. У этого метода третий параметр (*Отбор*) предназначен для фильтрации выборки. В этот параметр передается структура, ключ которой описывает имя поля, а значение структуры – значение отбора по этому полю:

```
Отбор = Новый Структура ("Регион", Регион);
ВыборкаКонтрагентов = Справочники.Контрагенты.Выбрать (, , Отбор);
Пока ВыборкаКонтрагентов.Следующий () Цикл
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = ВыборкаКонтрагентов.Наименование;
    Сообщение.Сообщить ();
КонецЦикла;
```

Обратите внимание на ограничения при применении отборов в методе *Выбрать()*. В качестве полей для отбора могут задаваться только поля *Код*, *Наименование* и реквизиты справочника, для которых в конфигураторе признак индексирования установлен в значении «Индексировать» или в значении «Индексировать с доп. упорядочиванием». Также важно, что структура может содержать только один элемент. Это означает, что решить задачу поиска, например, элемента справочника по нескольким реквизитам при помощи объектной модели доступа к данным нельзя.

**Пример 4.****Изменение значения реквизита для элемента справочника «Контрагенты»**

У элемента «Мосхлеб ОАО» справочника «Контрагенты» необходимо установить строковый реквизит *Город* в значение «Москва». Вначале для получения ссылки будет использован метод *НайтиПоНаименованию()* объекта типа *СправочникМенеджер.Контрагенты*. Затем из ссылки при помощи метода *ПолучитьОбъект()* будет получен объект типа *СправочникОбъект.Контрагенты*, который имеет метод *Записать()* для сохранения элемента справочника в базу данных:

```
СпрСсылка = Справочники.Контрагенты.НайтиПоНаименованию ("Мосхлеб ОАО");
СпрОбъект = СпрСсылка.ПолучитьОбъект ();
СпрОбъект.Город = "Москва";
Попытка
    СпрОбъект.Записать ();
Исключение
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "Не удалось записать объект "" + СпрОбъект + ""!";
    Сообщение.Сообщить ();
КонецПопытки;
```

При использовании объектной модели доступа к данным возможно как чтение данных, так и запись данных в базу. Важно понимать, что при этом на низком (системном) уровне методы инициируют тот или иной запрос к таблице справочника в базе данных.

Также при помощи объектной модели возможна модификация существующих элементов (заполнение реквизитов, установка пометки удаления и т.д.), а также физическое удаление элемента из базы. Таким образом, при использовании объектной модели доступен весь набор допустимых операций.

## Преимущества и ограничения объектного доступа

Как уже упоминалось, при объектной модели доступа возможно как чтение данных, так и запись. Тем не менее, на практике объектное чтение данных применяется достаточно редко. Это связано с наличием ограничений при чтении с использованием методов объектов встроенного языка.

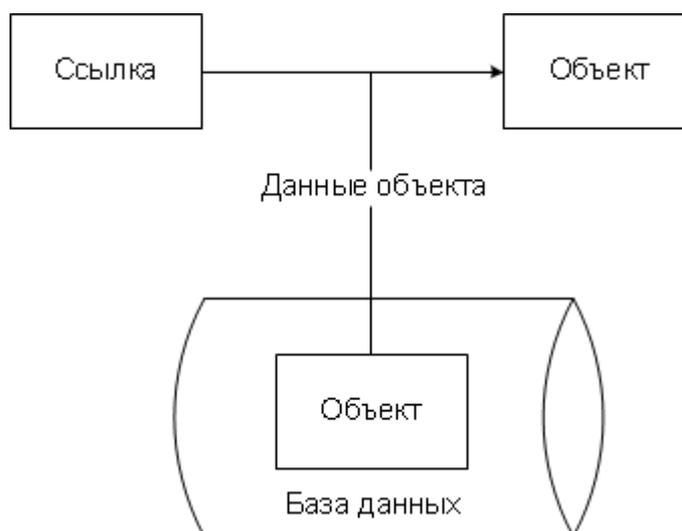
Во-первых, нет возможности наложить произвольные отборы на выбираемые данные. Так, при использовании отборов в методе *Выбрать()* в качестве полей для отбора могут задаваться только поля *Код*, *Наименование* и реквизиты справочника, для которых в конфигураторе признак индексирования установлен в значении «Индексировать» или в значении «Индексировать с доп. упорядочиванием». Кроме этого, структура отбора может содержать только один элемент. Такие ограничения делают чтение с использованием объектного доступа малоприменимым для широкого применения.

Во-вторых, при объектном доступе нет возможности выбирать данные из нескольких источников. К примеру, для получения данных по план-фактному анализу продаж необходимо сначала создать выборку из базы плановых данных, а затем – фактических, после этого объединить их в результирующий итог в виде, например, отчета. Получить все нужные данные за одно обращение к базе данных с использованием объектной техники, к сожалению, невозможно.

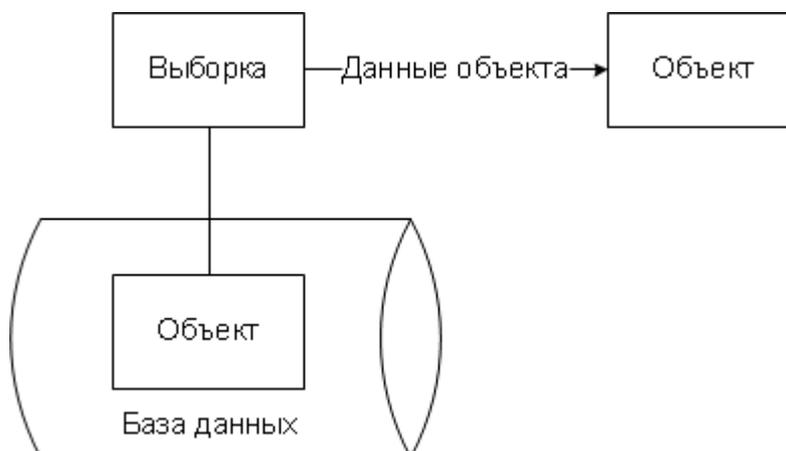
Между тем, объектное чтение вполне возможно использовать при простой выборке информации из базы данных.

Кроме этого, в платформе заложены механизмы оптимизации при массовой модификации элементов. Записать элемент в базу можно при использовании данных типа *Объект* (*справочникОбъект*, *ДокументОбъект* и т.д.). Значение такого типа можно получить двумя способами.

Во-первых, значение типа *СправочникОбъект* может быть получено из ссылки при вызове метода *ПолучитьОбъект()*. В этом случае будет создан экземпляр объекта и выполнено чтение данных из базы. При этом считываются значения реквизитов объекта и значения реквизитов всех его табличных частей.



Во-вторых, значение типа *СправочникОбъект* может быть получено из выборки (тип данных *СправочникВыборка*) при вызове метода с аналогичным названием – *ПолучитьОбъект()*. В этом случае также будет создан экземпляр объекта, однако чтение из базы данных выполняться не будет, т. к. выборка считывает из базы все данные объектов, и данные созданного экземпляра объекта будут заполнены непосредственно из выборки.

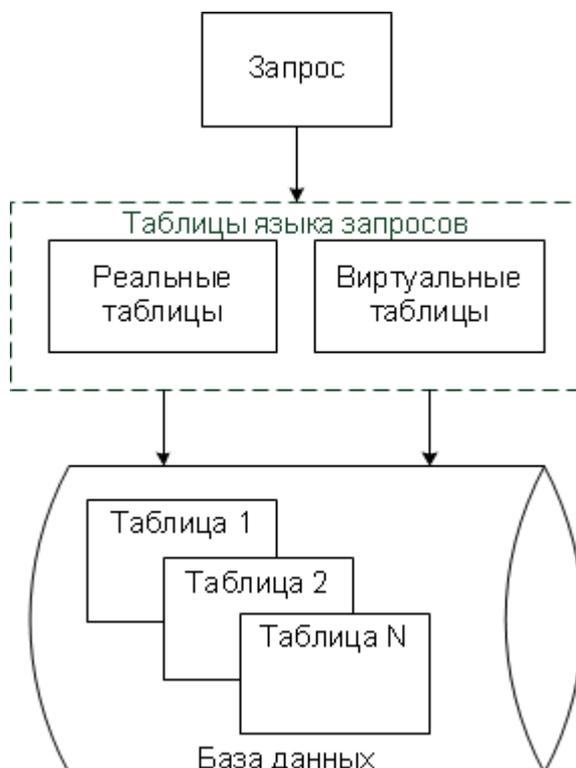


Как видно, при использовании получения объекта из выборки не будет проводиться дополнительное чтение из базы, поэтому такой способ является более производительным при массовой модификации элементов.

## Табличная модель доступа к данным

При табличной модели доступа к данным информационная база представляется в виде набора связанных двумерных («плоских») таблиц. Каждая таблица базы данных состоит из строк и столбцов. А каждая строка (запись) таблицы представляет собой ряд значений, содержащихся в ее столбцах (полях).

В табличной модели доступна только операция чтения данных. Для этого разработчик формирует запрос на специальном языке запросов системы «1С:Предприятие 8». При выполнении такого запроса платформа преобразует его текст в форму, понятную серверу баз данных. Именно при помощи такого запроса необходимые данные извлекаются из базы. Схематически это может быть изображено следующим образом:



Подробно физические (реальные) и виртуальные таблицы рассматриваются ниже.

Операции записи, модификации и удаления данных недоступны в табличной модели доступа к данным. Это связано со сложной структурой хранения данных, когда один объект метаданных физически хранится в базе данных в виде нескольких таблиц.

Рассмотрим для примера регистр накопления. В базе данных записи регистра хранятся в двух таблицах – в основной таблице движений и вспомогательной таблице итогов. При записи в таблицу движений регистра платформа самостоятельно изменяет записи в таблице итогов. При этом таблица итогов недоступна для непосредственного обращения к ней разработчика. Только платформа полностью знает, как именно будут изменены данные таблицы итогов. Тем самым на уровне платформы обеспечивается целостность данных. Т.е. когда разработчик в программном коде использует объектную модель доступа, платформа самостоятельно учитывает существующие взаимосвязи между различными таблицами одного объекта. Разработчику не нужно держать в голове все эти зависимости. Кроме того, программист может забыть синхронно изменить данные в одной из физических таблиц объекта, тем самым нарушив целостность данных. Именно поэтому операции записи, модификации и удаления данных недоступны в табличной модели.

Использование табличного метода при чтении данных является более приоритетным, поскольку СУБД имеют встроенные средства оптимизации (поиск оптимального плана выполнения запроса на основании некоторых сведений – потенциальное число строк, извлекаемое из каждой таблицы, получаемое из статистики, наличие индексов, наличие сохраненного плана запроса в кэше и т.д.).

Также обратите внимание, что при табличном подходе возможно извлечение данных из нескольких таблиц за одно обращение к базе данных. Это позволяет получать, например, все необходимые для отчета данные одним запросом, что уменьшает объем необходимого программного кода для обработки полученных данных, а также увеличивает производительность системы.

Стоит также остановиться на еще одном важном отличии рассмотренных моделей доступа к данным. Если в выборке, полученной при помощи объектной модели, присутствуют данные, на которые у пользователя нет прав доступа, то возникнет ошибка. А при использовании табличной модели есть возможность получить только те данные, на которые есть права доступа, с помощью использования в запросе ключевого слова РАЗРЕШЕННЫЕ.

## Таблицы объектов метаданных

Каждому объекту конфигурации соответствует одна или несколько таблиц информационной базы в зависимости от вида объекта и его структуры.

Таблицы информационной базы, определяемые структурой метаданных, имеют имена, создаваемые по одному и тому же правилу. Имя таблицы начинается с буквенного префикса, который обозначает принадлежность таблицы определенному виду объектов. После префикса следует номер (далее обозначается <n>), который позволяет различать таблицы объектов одинакового вида. Таблицы, содержащие табличные части объекта, имеют тот же номер <n>, что и основная таблица объекта. После префикса и номера, определяющих принадлежность к некоторому объекту конфигурации, идет префикс \_VT, указывающий, что это таблица табличной части, и номер (далее обозначается <k>), позволяющий различить несколько табличных частей одного и того же объекта конфигурации.

Рассмотрим, какие таблицы существуют для каждого из прикладных объектов метаданных. При этом, конечно же, не будем использовать привязку к какой-либо конкретной конфигурации.

### Константы

\_Consts – таблица констант.

Все значения констант, определенных в конфигурации, хранятся в одной таблице.

\_ConstsChngR – таблица регистрации изменений констант. Создается, если хотя бы одна константа участвует хотя бы в одном плане обмена.

## Планы обмена

\_Node<n> – основная таблица плана обмена.

\_Node<n>\_VT<k> – табличная часть плана обмена.

## Справочники

\_Reference<n> – основная таблица справочника.

\_Reference<n>\_VT<k> – табличная часть справочника.

\_ReferenceChngR<n> – таблица регистрации изменений справочника. Создается, если справочник участвует хотя бы в одном плане обмена.

## Документы

\_Document<n> – основная таблица документа.

\_Document<n>\_VT<k> – табличная часть документа.

\_DocumentChngR<n> – таблица регистрации изменений документа. Создается для каждого объекта метаданных *Документ*, который участвует хотя бы в одном плане обмена.

## Последовательности

\_Seq<n> – таблица регистрации документов в последовательности.

\_SeqB<n> – таблица границ последовательности.

Эти таблицы имеют сквозную нумерацию.

\_SeqChngR<n> – таблица регистрации изменений последовательности. Создается для каждой последовательности, которая участвует хотя бы в одном плане обмена.

## Журналы документов

\_DocumentJournal<n> – таблица журнала документов.

## Перечисления

\_Enum<n> – таблица перечисления.

## Планы видов характеристик

\_Chrc<n> – основная таблица плана видов характеристик.

\_Chrc<n>\_VT<k> – табличная часть плана видов характеристик.

\_ChrcChngR<n> – таблица регистрации изменений плана видов характеристик. Создается, если план видов характеристик участвует хотя бы в одном плане обмена.

## Планы счетов

\_Acc<n> – основная таблица плана счетов.

\_Acc<n>\_ExtDim<k> – таблица видов субконто плана счетов. Эта таблица создается в том случае, если максимальное количество субконто больше нуля.

\_Acc<n>\_VT<k> – табличная часть плана счетов.

\_AccChngR<n> – таблица регистрации изменений плана счетов. Создается, если план счетов участвует хотя бы в одном плане обмена.

## Планы видов расчета

\_CKind<n> – основная таблица плана видов расчета.

\_CKind<n>\_BaseCK – таблица базовых видов расчета. Эта таблица создается, если у плана видов расчета свойство *Зависимость от базы* имеет значение, отличное от *Не зависит*.

\_CKind<n>\_DisplacedCK – таблица вытесняемых видов расчета.

Эта таблица создается в случае, если для плана видов расчета установлен флаг *Использует период действия*.

\_CKind<n>\_LeadingCK – таблица ведущих видов расчета.

\_CKindDN<n> – вспомогательная таблица для порядка вытеснения.

Эта таблица создается в случае, если у плана видов расчета установлен флаг *Использует период действия*.

\_CKind<n>\_VT<k> – табличная часть плана видов расчета.

Все таблицы плана видов расчета имеют один и тот же номер <n>, за исключением вспомогательной таблицы для порядка вытеснения. Для нее используется сквозная нумерация.

\_CKindChngR<n> – таблица регистрации изменений плана видов расчета. Создается, если план видов расчета участвует хотя бы в одном плане обмена.

## Регистры сведений

\_InfoRg<n> – таблица движений регистра сведений.

\_InfoRgChngR<n> – таблица регистрации изменений регистра сведений. Создается, если регистр сведений участвует хотя бы в одном плане обмена.

## Регистры накопления

\_AccumRg<n> – таблица движений регистра накопления.

\_AccumRgT<n> – таблица итогов регистра накопления. Эта таблица создается в случае, если регистр накопления поддерживает остатки.

\_AccumRgTn<n> – таблица оборотов регистра накопления. Эта таблица создается, если регистр поддерживает обороты.

\_AccumRgOpt – таблица настроек хранения итогов регистров накопления. Эта таблица создается одна на все регистры накопления.

\_AccumRgAgg<n> – таблица агрегатов регистра накопления.

\_AccumRgAggOpt – таблица опций сети агрегатов.

\_AccumRgSt<n> – таблица статистики регистра накопления.

\_AccumRgBf<n> – таблица буфера новых оборотов регистра накопления.

\_AccumRgDI<n> – таблица новых оборотов регистра накопления.

\_AccumRgAggDims – таблица кодов измерений регистра накопления.

\_AccumRgAggGrid – таблица сети агрегатов.

Эти таблицы имеют сквозную нумерацию.

\_AccumRgChngR<n> – таблица регистрации изменений регистра накопления. Создается, если регистр накопления участвует хотя бы в одном плане обмена.

## Регистры бухгалтерии

\_AccRg<n> – таблица движений регистра бухгалтерии.

\_AccRgED<n> – таблица значений субконто регистра бухгалтерии. Эта таблица создается в том случае, если регистр бухгалтерии ссылается на план счетов, у которого максимальное количество субконто больше нуля.

\_AccRgAT0<n> – таблица остатков и оборотов по счетам и субсчетам.

\_AccRgAT<i><n> – таблица остатков и оборотов по счетам, субсчетам и субконто.

Эти таблицы создаются в том случае, если регистр бухгалтерии ссылается на план счетов, у которого максимальное количество субконто больше нуля. Номер *i* изменяется от 1 до максимального количества субконто.

\_AccRgCT<n> – таблица итогов оборотов между счетами. Эта таблица создается только для регистра бухгалтерии, поддерживающего корреспонденцию.

\_AccRgOpt – таблица настроек хранения итогов. Эта таблица создается одна на все регистры бухгалтерии.

Все таблицы регистра бухгалтерии имеют сквозную нумерацию.

\_AccRgChngR<n> – таблица регистрации изменений регистра бухгалтерии. Создается, если регистр бухгалтерии участвует хотя бы в одном плане обмена.

## Регистры расчета

\_CRg<n> – таблица движений регистра расчета.

\_CRgActP<n> – таблица фактического периода действия. Эта таблица создается в том случае, если у регистра расчета установлен флаг *Период действия*.

\_CRgRecalc<n> – таблица перерасчета регистра расчета.

Все таблицы регистра расчета имеют сквозную нумерацию.

\_CRgChngR<n> – таблица регистрации изменений регистра расчета. Создается для каждого регистра расчета, участвующего хотя бы в одном плане обмена.

\_CRgRecalcChngR<n> – таблица регистрации изменений перерасчета. Создается, если перерасчет участвует хотя бы в одном плане обмена.

## Бизнес-процессы

\_BPr<n> – основная таблица бизнес-процесса.

\_BPr<n>\_VT<k> – табличная часть бизнес-процесса.

\_BPRPoints<n> – таблица точек маршрута бизнес-процесса.

Таблицы, содержащие табличные части бизнес-процесса, имеют тот же номер <n>, что и основная таблица бизнес-процесса, однако для таблицы точек маршрута бизнес-процесса используется сквозная нумерация.

\_BPrChngR<n> – таблица регистрации изменений бизнес-процесса. Создается для каждого бизнес-процесса, участвующего хотя бы в одном плане обмена.

## Задачи

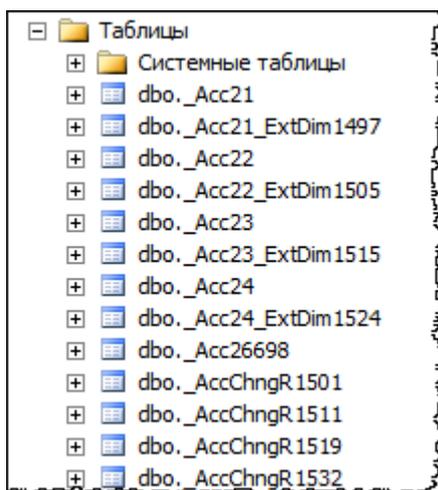
\_Task<n> – основная таблица задачи.

\_Task<n>\_VT<k> – табличная часть задачи.

Таблицы, содержащие табличные части задачи, имеют тот же номер <n>, что и основная таблица задачи.

\_TaskChngRec<n> – таблица регистрации изменений в задачах. Создается для каждого объекта метаданных *Задача*, который участвует хотя бы в одном плане обмена.

Именно с такими именами можно столкнуться, если открыть на просмотр список таблиц базы данных, например, при помощи Microsoft SQL Server Management Studio:



Такие имена таблиц можно увидеть при анализе запросов, непосредственно выполняющихся на сервере баз данных. При выполнении работ по оптимизации запросов при помощи инструмента Microsoft SQL Server Profiler необходимо знать, к какому объекту метаданных относится та или иная таблица в базе.

Кроме этого, в технологическом журнале тексты запросов регистрируются как на языке запросов «1С:Предприятие 8», так и в терминах СУБД. Также тексты запросов в этих двух видах можно получить при помощи специализированной обработки – «Консоли запросов».

Еще в технологическом журнале можно получить план выполнения запроса. В этой ситуации тоже необходимо знать, таблицы каких объектов в нем фигурируют.

Также непосредственное обращение к таблицам может потребоваться в некоторых случаях при восстановлении работоспособности информационной базы, например, если зайти в конфигуратор невозможно из-за ошибки, а также при групповом изменении или удалении данных при помощи прямых запросов к базе данных.

Однако необходимо помнить, что согласно лицензионному соглашению нельзя обращаться к данным информационной базы напрямую, минуя уровень объектов работы с данными «1С:Предприятие 8», например, при помощи средств СУБД или внешних компонент, которые реализуют прямой доступ к СУБД.

Для того чтобы узнать, для чего предназначена та или иная таблица в конкретной базе данных, в платформе «1С:Предприятие 8» существует метод глобального контекста.

[Получить Структуру Хранения Базы Данных \(\)](#)

Эта функция возвращает таблицу значений с описаниями структуры таблиц, индексов и полей базы данных в терминах модели объектов метаданных или используемой СУБД:

ИмяТаблицыХранения	ИмяТаблицы	Метаданные
"Reference158"	"Справочник.Организации"	"Справочник.Организации"
"ReferenceChngR301"	"Справочник.Организации.Изменения"	"Справочник.Организации"
"Reference193"	"Справочник.ВходящиеПисьма"	"Справочник.ВходящиеПисьма"
"Reference194"	"Справочник.ИсходящиеПисьма"	"Справочник.ИсходящиеПисьма"
"Reference195"	"Справочник.ПочтовыеВложения"	"Справочник.ПочтовыеВложения"
"Reference289"	"Справочник.Пользователи"	"Справочник.Пользователи"
"ReferenceChngR303"	"Справочник.Пользователи.Изменения"	"Справочник.Пользователи"
"Document17"	"Документ.ПриходТовара"	"Документ.ПриходТовара"
"Document17.VT43"	"Документ.ПриходТовара.Товары"	"Документ.ПриходТовара.ТабличнаяЧасть.Товары"
"Document18"	"Документ.РасходТовара"	"Документ.РасходТовара"
"Document18.VT53"	"Документ.РасходТовара.Товары"	"Документ.РасходТовара.ТабличнаяЧасть.Товары"
"Document19"	"Документ.Оплата"	"Документ.Оплата"
"Document20"	"Документ.ПоступлениеДенег"	"Документ.ПоступлениеДенег"
"Document131"	"Документ.ОперацияПоУчетуТоваров"	"Документ.ОперацияПоУчетуТоваров"
"Document290"	"Документ.Заказ"	"Документ.Заказ"
"Document290.VT311"	"Документ.Заказ.Товары"	"Документ.Заказ.ТабличнаяЧасть.Товары"
"DocumentChngR317"	"Документ.Заказ.Изменения"	"Документ.Заказ"
"DocumentJournal103"	"ЖурналДокументов.ФинансовыеДокументы"	"ЖурналДокументов.ФинансовыеДокументы"
"DocumentJournal318"	"ЖурналДокументов.ДокументыПродаж"	"ЖурналДокументов.ДокументыПродаж"
"Enum116"	"Перечисление.ВидыТоваров"	"Перечисление.ВидыТоваров"
"Enum190"	"Перечисление.НазначениеХарактеристик"	"Перечисление.НазначениеХарактеристик"

ИмяПоляХранения	ИмяПоля	Метаданные
"ID"	"Ссылка"	""
"Version"	"ВерсияДанных"	""
"Marked"	"ПометкаУдаления"	""
"PredefinedID"	"ИмяПредопределенныхДанных"	""
"ParentID"	"Родитель"	""
"Folder"	"ЭтоГруппа"	""
"Code"	"Код"	""
"Description"	"Наименование"	""
"Fld26"	"Регион"	"Справочник.Контрагенты.Реквизит.Регион"
"Fld27"	"Индекс"	"Справочник.Контрагенты.Реквизит.Индекс"
"Fld28"	"Страна"	"Справочник.Контрагенты.Реквизит.Страна"
"Fld29"	"Город"	"Справочник.Контрагенты.Реквизит.Город"
"Fld30"	"Улица"	"Справочник.Контрагенты.Реквизит.Улица"
"Fld31"	"Дом"	"Справочник.Контрагенты.Реквизит.Дом"
"Fld32"	"Телефон"	"Справочник.Контрагенты.Реквизит.Телефон"

Отметим некоторые особенности хранения данных в базе.

При использовании информационной базы IBM DB2 префиксы псевдонимов таблиц начинаются не с символа подчеркивания, а сразу с буквенной части.

При режиме совместимости старше, чем Версия 8.2.13, изменен режим хранения констант и настроек регистров накопления. Для каждого объекта используется своя таблица базы данных. При включении режима совместимости (в значении Версия 8.2.13 или Версия 8.1) выполняется обратная конвертация для обеспечения возможности запуска прикладного решения с помощью версии 8.2.13. В платформе 8.3.1 для периодического регистра сведений в базе данных появилась новая таблица – физическая таблица итогов (по аналогии с таблицей итогов регистра накопления). Эта таблица используется для получения срезов первых или последних при выполнении всех следующих условий:

- Разрешено использование итогов (в метаданных)
- Установлено использование итогов (в режиме «1С:Предприятие 8»)
- Срез получается без указания даты
- В условиях получения среза используются только измерения и разделители
- В ограничениях доступа к данным для регистра используются только измерения и разделители.

Тем самым повышена скорость получения срезов первых или последних на текущий момент времени.

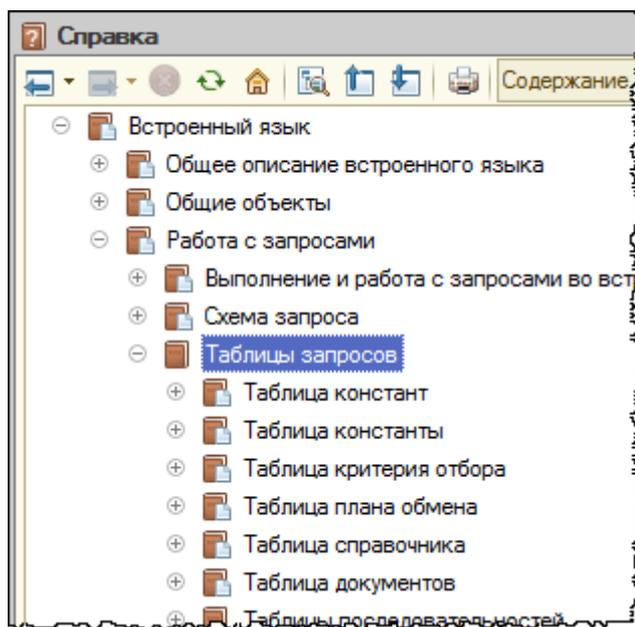
## Физические и виртуальные таблицы

Важно понимать, что непосредственного доступа к физическим таблицам, в которых хранятся данные (именно они были подробно описаны в предыдущем разделе), из «1С:Предприятие 8» получить нельзя.

Это связано с тем, что «1С:Предприятие 8» может работать с четырьмя различными СУБД (Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database), а также с файловой информационной базой собственного формата. При этом каждый вариант имеет свою специфику, а текст запроса должен быть универсальным, а также возвращать одинаковый результат в любом случае. Поэтому платформа транслирует текст запроса на языке системы «1С:Предприятие 8» в запрос в терминах, понятных конкретной СУБД, с которой происходит работа.

Также важно, что язык запросов «1С:Предприятие 8» оперирует «предметными» сущностями (например, *Справочник.Контрагенты*), а в базе данных физически хранятся таблицы с совершенно другими именами (например, *\_Reference54*). Т.е. можно сказать, что с помощью запросов обращаемся не к данным непосредственно, а путем обращения к **таблицам языка запроса**.

Подробное описание этих таблиц приведено в справке в разделе *Встроенный язык – Работа с запросами – Таблицы запросов*:



Итак, все таблицы, к которым можно обратиться с помощью языка запросов, делят на физические и виртуальные таблицы.

Рассмотрим их подробнее.

## Физические таблицы

Физические (реальные) таблицы на самом деле присутствуют в базе данных. Так, например, таблица *Справочник.Контрагенты*, соответствующая справочнику «Контрагенты» в дереве объектов метаданных, является физической. Однако в базе данных она имеет абсолютно другое имя (например, *\_Reference54*).

Также важно понимать, что поля (реквизиты) физической таблицы также имеют «предметное» наименование (например, стандартные реквизиты имеют название *ПометкаУдаления*, *Предопределенный*), в то время как в базе данных эти же поля имеют другое название (*\_Marked*, *\_IsMetadata*). Но в общем, несмотря на эти различия, физическая таблица схожа со своим аналогом, непосредственно хранящимся в базе данных, а также имеет такое же количество записей.

## Виртуальные таблицы

Виртуальных таблиц на самом деле в базе данных не существует. Они сделаны для удобства разработчиков. Виртуальные таблицы формируются в момент выполнения запроса на основе реальных таблиц базы данных. Например, виртуальная таблица *РегистрСведений.КурсыВалют.СрезПоследних* формируется на основе таблицы регистра сведений *КурсыВалют*.

Виртуальные таблицы позволяют получить данные без необходимости составления сложных запросов, не задумываясь о том, как на самом деле хранится в базе данных тот или иной объект.

Рассмотрим для примера получение данных на определенную дату из виртуальной таблицы *РегистрНакопления.ТоварныеЗапасы.Остатки*. Схематически выборка данных из виртуальной таблицы приведена ниже:



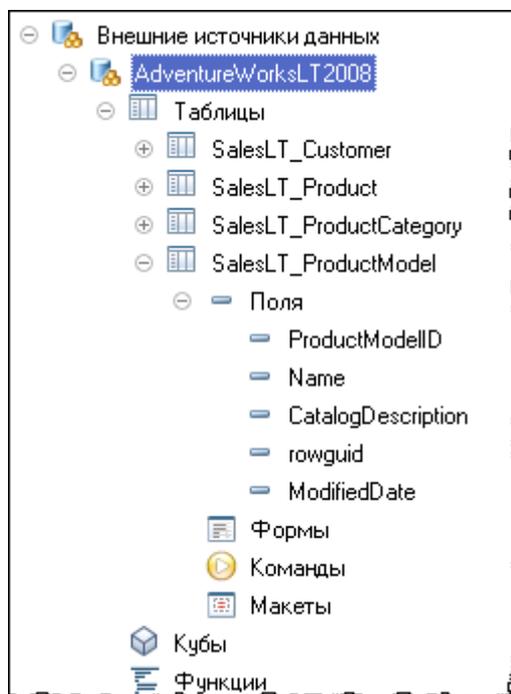
Как отмечалось ранее, для регистра накопления в базе присутствует таблица движений (записей) и таблица итогов. Платформа сначала получает текущие остатки регистра (остатки на текущую дату), которые хранятся в таблице итогов на дату 01.11.3999 00:00:00, затем корректирует эти остатки в соответствии с записями из таблицы движений за период с даты получения остатков по дату текущих остатков. Если бы в платформе не существовало виртуальных таблиц, разработчикам пришлось бы каждый раз получать итоги самостоятельно, при помощи объединений запросов и группировок. Поэтому теперь разработчик может полностью сосредоточиться на бизнес-логике, а не на алгоритме получения остатков из регистра накопления.

Таким образом, виртуальная таблица представляет собой некий интерфейс для удобства разработчика («черный ящик»), на входе которого указывается набор параметров – условий для получения данных из базы. Во время выполнения эта конструкция будет преобразована в конкретный запрос к базе данных в зависимости от набора переданных параметров.

## Внешние источники данных

Часто при работе системы на базе «1С:Предприятие 8» необходимо получать информацию из внешних реляционных баз данных или OLAP-систем (Microsoft Analysis Services, Oracle

Essbase, IBM InfoSphere Warehouse). Эта информация может использоваться для построения отчетов или как основа для проведения некоторых расчетов. Для таких задач в системе существует объект метаданных *ВнешниеИсточникиДанных*.



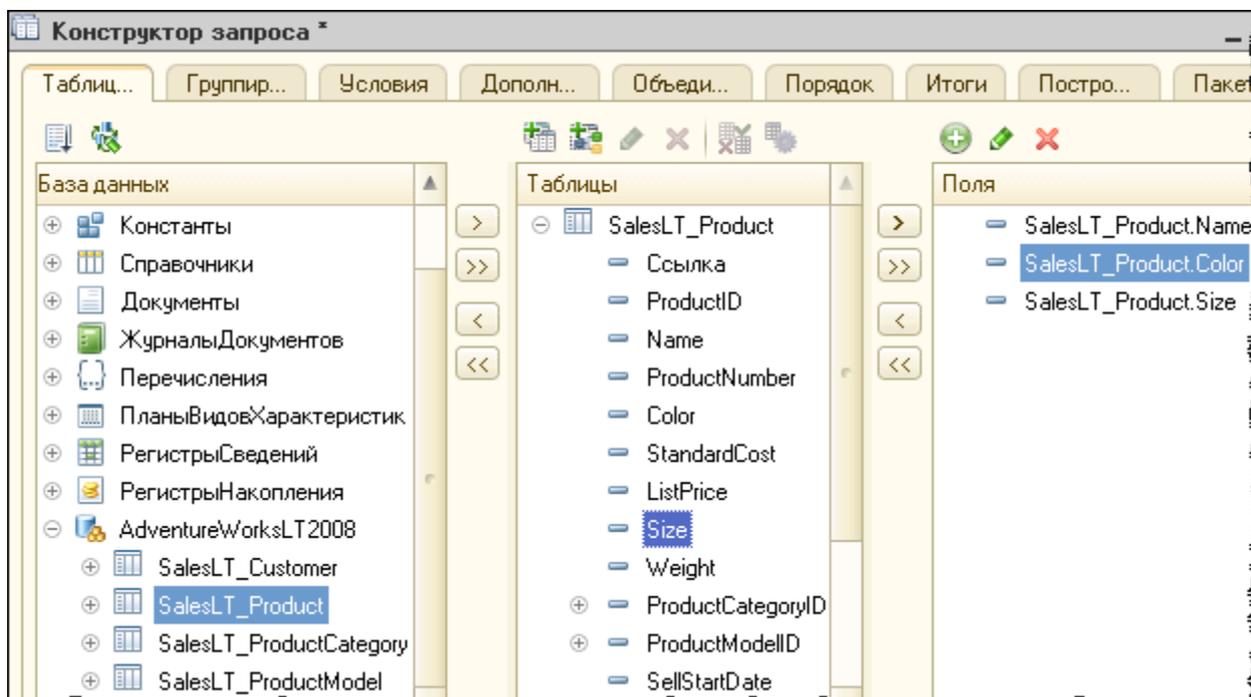
Для получения данных из внешних источников данных используется механизм ODBC.

При помощи внешних источников с данными, не хранящимися непосредственно в базе «1С:Предприятие 8», можно работать внутри конфигурации так же, как если бы данные хранились в самой базе.

ProductID	Name	ProductNumber	Color
680	HL Road Frame - Black, 58	FR-R92B-58	Black
706	HL Road Frame - Red, 58	FR-R92R-58	Red
707	Sport-100 Helmet, Red	HL-U509-R	Red
708	Sport-100 Helmet, Black	HL-U509	Black
709	Mountain Bike Socks, M	SO-B909-M	White

Объект, который описывает внешний источник данных, подключенный к базе данных, состоит из таблиц, а каждая таблица состоит из полей. Этот объект можно использовать в

качестве источника данных в запросе, как и другие таблицы самой информационной базы. Кроме этого, внешний источник данных можно использовать как источник в системе компоновки данных, в динамических списках.



При работе внешнего источника с Microsoft SQL Server, IBM DB2, PostgreSQL и Oracle Database реализованы все возможности языка запросов «1С:Предприятие 8». При использовании других СУБД функционал ограничен возможностями используемого драйвера ODBC.

## Вопросы и ответы в Мастер-группе



Если у Вас возникли вопросы по материалам текущего модуля курса, Вы можете [задать их на странице сайта](#):

[Перейти к вопросам](#)

На этой же странице Вы можете просмотреть ответы на вопросы других участников курса и почерпнуть новые знания. Данная возможность доступна только участникам платной версии тренинга.

## Модуль 2. Синтаксис текста запроса

### Язык запросов

Механизм запросов позволяет получить необходимые данные из информационной базы «1С:Предприятие 8». Для исполнения запроса необходимо написать текст запроса. Текст запроса представляет собой набор инструкций для извлечения данных из базы. В тексте запроса указывается источник данных – одна таблицы базы или несколько таблиц, указываются поля, которые необходимо выбрать, также можно описать условия, которым должны отвечать результирующие данные, правила группировки, сортировки и т.д. Затем полученный результат можно обработать при помощи встроенного языка, использовать далее в разрабатываемых алгоритмах конфигурации, для построения отчетов и т.д.

Инструкция составляется на специальном языке (языке запросов) и состоит из отдельных частей – секций, предложений, слов, функций и комментариев.

Важно понимать, что при помощи запросов можно только получать информацию из базы. Записать, удалить или модифицировать данные непосредственно с помощью запросов нельзя. Для этого следует воспользоваться объектной моделью доступа к данным.

Однако при помощи запросов зачастую формируется выборка для последующего изменения данных.

Например, запросом из базы выбираются данные, которые используются для заполнения табличной части документа, или запросом получается список элементов справочника, которые впоследствии будут скопированы либо удалены. Таким образом, сам по себе запрос в платформе «1С:Предприятие 8» данные изменить не может, но часто используется с целью получения данных для дальнейшей модификации.

Также важно, что текст запроса на языке «1С:Предприятие 8» транслируется при выполнении в термины, понятные определенной СУБД – будут подставлены реальные имена таблиц в базе, виртуальные таблицы будут преобразованы в конкретный запрос к базе данных.

## Сравнение языка запросов с SQL

Язык запросов достаточно похож на классический язык SQL, однако надо понимать, что классический язык запросов – единственный способ взаимодействия с базой данных. Платформа «1С:Предприятие 8» нацелена на решение задач автоматизации предприятий, поэтому и язык запросов обладает отличиями, имеет собственные особенности. Рассмотрим эти моменты подробнее.

Язык запросов «1С:Предприятие 8» оперирует «предметными» сущностями (например, *Справочник.Товары*), а в базе данных физически хранятся таблицы с совершенно другими

именами (например, \_Reference7). Запрос в предметных терминах переводится в запрос к таблицам базы данных, затем выполняется на сервере СУБД. Например, запрос на языке запросов «1С:Предприятие 8» выглядит следующим образом:

ВЫБРАТЬ

Товары.Ссылка,  
Товары.Наименование

ИЗ

Справочник.Товары КАК Товары

На сервер СУБД будет передан не приведенный выше текст, а запрос, в котором используются имена реально хранящихся в базе данных таблиц:

SELECT

T1.\_IDRRef,  
T1.\_Description

FROM \_Reference7 T1

Именно этот запрос и будет выполнен на сервере СУБД. Платформа «1С:Предприятие 8» поддерживает также работу с файловой базой данных собственного формата. Файловая СУБД является частью платформы. При этом все данные информационной базы хранятся в одном файле 1Сv8.1CD. А для работы с базой не требуется дополнительное программное обеспечение, необходима только операционная система и платформа «1С:Предприятие 8». При этом разработчику не нужно беспокоиться о том, в каком режиме будут работать пользователи, поскольку текст запроса будет одинаковым как в файловом, так и в клиент-серверном варианте.

Также нужно помнить, что согласно лицензионному соглашению можно обращаться к данным информационной базы только при помощи объектов «1С:Предприятие 8», специально предназначенных для работы с данными (запросы, справочники, документы и т.д.). Нельзя обращаться к данным информационной базы напрямую, минуя уровень объектов работы с данными «1С:Предприятие 8», например, при помощи средств СУБД или при помощи внешних компонент, которые реализуют прямой доступ к СУБД.

Язык запросов имеет также англоязычный вариант написания.

В языке запросов нельзя использовать запрос в качестве выходного поля. Т.е. конструкции следующего вида недопустимы:

SELECT

...,

(SELECT ... FROM ...),

...

FROM ...

В языке запросов «1С:Предприятие 8» нельзя использовать хранимые процедуры (stored procedure) и представления (view), которые существуют в Microsoft SQL Server.

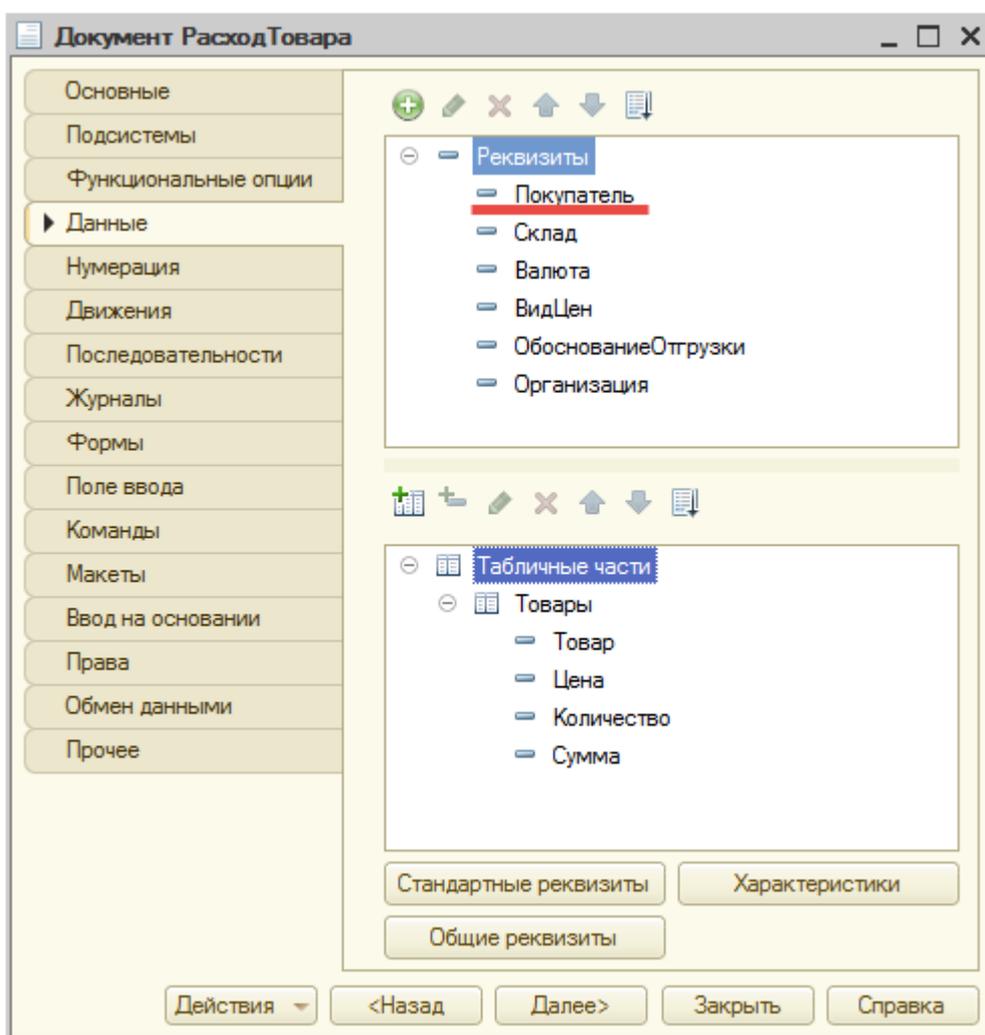
Также важный недостаток – в языке запросов нельзя выполнять преобразование строки в число, хотя в SQL такое выражение абсолютно корректно:

```
CAST('10' AS INT)
```

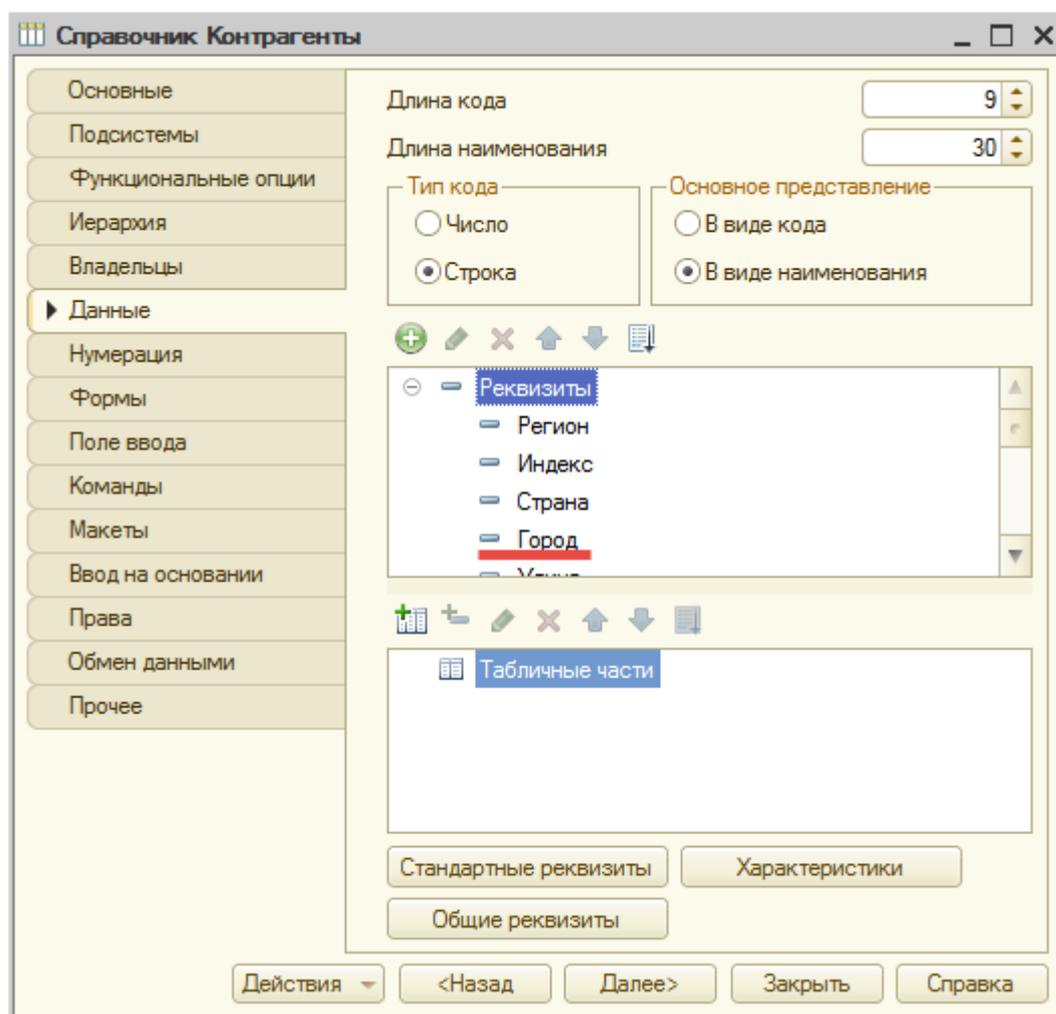
Запрос в «1С:Предприятие 8» предназначен для решения определенной задачи – это запрос к базе данных из «1С:Предприятие». Этим обусловлено наличие особенностей, которых не существует в классическом SQL.

Язык запросов поддерживает разыменовывание полей. Это значит, что возможно обращаться не только к полям исходных таблиц запроса, но и к полям таблицы, на которую ссылается поле исходной таблицы запроса, если оно имеет ссылочный тип.

Рассмотрим пример. В конфигурации существует документ *РасходТоваров*, у которого есть реквизит Покупатель с типом *СправочникСсылка.Контрагенты*:



Далее, у справочника «Контрагенты» есть строковый реквизит *Город*:



Тогда если в запросе используется таблица документа *РасходТовара*, то допустимо обратиться к полю *Город* реквизита *Покупатель*. Имена полей при этом пишутся через точку, т. е. *РасходТовара.Покупатель.Город*.

Применение такой конструкции позволяет улучшить наглядность и понятность текста запроса. С технической стороны на системном уровне при выполнении такого запроса в текст будет добавлена таблица справочника, и между таблицами документа и справочника будет установлена связь.

Язык запросов поддерживает обращения к вложенным табличным частям объектов и как к отдельным таблицам, и как к полям одной таблицы.

Например, в конфигурации есть документ *РасходТовара* с табличной частью *Товары*. При первом варианте обращения можно выбрать запросом записи непосредственно из таблицы, содержащей данные табличной части.

В этом случае будет получена такая таблица:

Документ	Товар	Количество	Сумма (руб.)
Продажа 13 от 02.02.2014	Сапоги	2	7 000,00
Продажа 13 от 02.02.2014	Ботинки	1	2 000,00
Продажа 28 от 17.02.2014	Сапоги	1	3 500,00

При втором варианте обращения можно считать запись из основной таблицы объекта (т.е. документа), тогда значением поля *Товары* будет вложенная таблица, содержащая все записи табличной части этого объекта. В этом случае результат запроса схематично будет выглядеть следующим образом:

Документ	Товары		
Продажа 13 от 02.02.2014	<b>Товар</b>	<b>Количество</b>	<b>Сумма (руб.)</b>
	Сапоги	2	7 000,00
	Ботинки	1	2 000,00
Продажа 28 от 17.02.2014	<b>Товар</b>	<b>Количество</b>	<b>Сумма (руб.)</b>
	Сапоги	1	3 500,00

Приведенная таблица демонстрирует, что в поле *Товары* находится не одно значение, а вложенная таблица, которая содержит строки табличной части документа.

Язык запросов позволяет формировать итоги. Использование итогов в запросе включает в результат дополнительные строки с рассчитанными агрегированными данными (например, сумма, максимум или минимум по определенной группировке).

Для приведенных выше данных результат запроса с итогами (суммарным количеством и стоимостью по товарам) будет выглядеть следующим образом:

Документ	Товар	Количество	Сумма (руб.)
----------	-------	------------	--------------

	<b>Сапоги</b>	<b>3</b>	<b>10 500,00</b>
Продажа 13 от 02.02.2014	Сапоги	2	7 000,00
Продажа 28 от 17.02.2014	Сапоги	1	3 500,00
	<b>Ботинки</b>	<b>1</b>	<b>2 000,00</b>
Продажа 13 от 02.02.2014	Ботинки	1	2 000,00

Причем, итоги и подитоги могут формироваться с учетом группировки и иерархии, а обход уровней итогов при помощи встроенного языка может выполняться в произвольном порядке с подведением подитогов. Если справочник «Товары» иерархический, то можно получить итоги с учетом существующих папок (групп) в этом справочнике. Так, товары «Сапоги» и «Ботинки» расположены в папке *Обувь*. Тогда итоги по иерархии будут выглядеть следующим образом:

Документ	Товар	Количество	Сумма(руб.)
	<b>Обувь</b>	<b>4</b>	<b>12 500,00</b>
	<b>Сапоги</b>	<b>3</b>	<b>10 500,00</b>
Продажа 13 от 02.02.2014	Сапоги	2	7 000,00
Продажа 28 от 17.02.2014	Сапоги	1	3 500,00
	<b>Ботинки</b>	<b>1</b>	<b>2 000,00</b>
Продажа 13 от 02.02.2014	Ботинки	1	2 000,00

Также язык запросов поддерживает виртуальные таблицы, которые упрощают разработку, а также повышают читаемость программного кода.

## Структура запроса

Текст запроса состоит из нескольких частей – разделов:

- Описание запроса
- Объединение запросов
- Сортировка результатов
- Автоупорядочивание
- Описание итогов.

Поскольку при помощи запросов можно только получать данные из базы, то текст запроса начинается с ключевого слова **ВЫБРАТЬ**.

Текст запроса обязательно должен содержать только первую часть – описание запроса. Все остальные части присутствуют в запросе только в случае необходимости. Раздел описания запроса содержит сведения о:

- Полях выборки
- Используемых таблицах и соединениях
- Условиях
- Группировках.

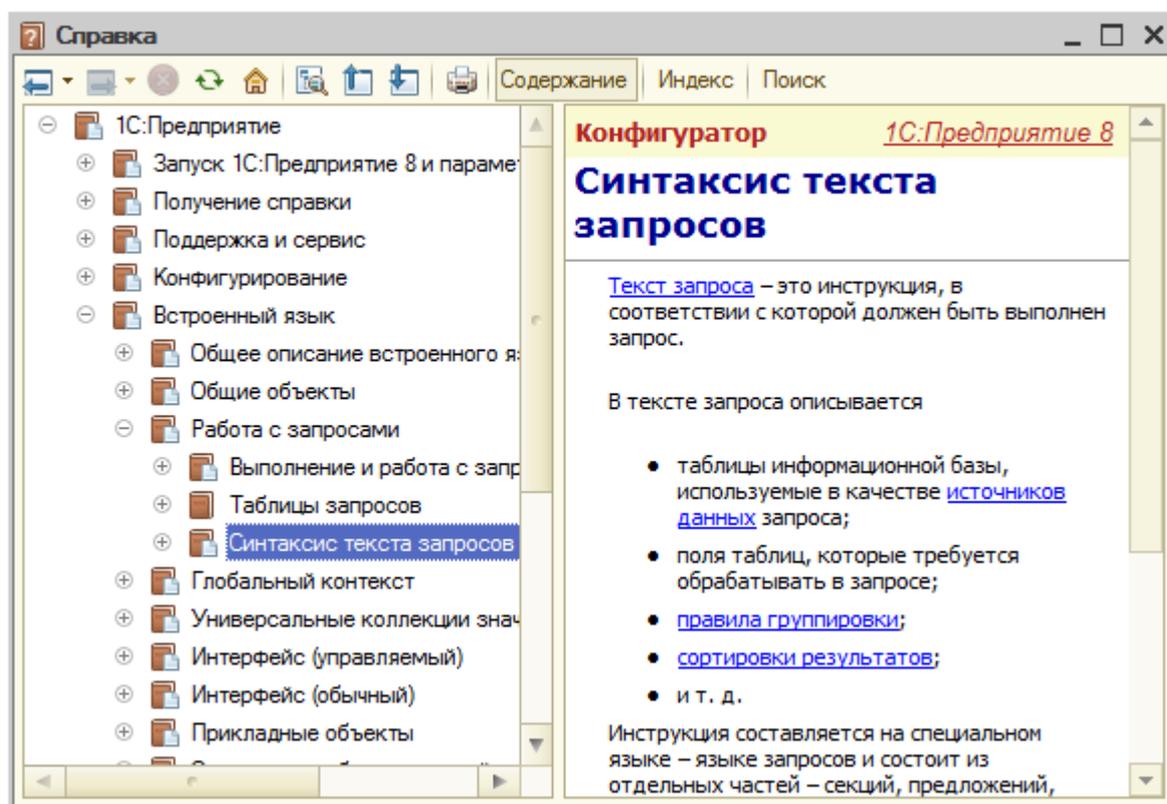
Причем в описании запроса обязательно должны присутствовать только выбираемые поля. Остальные пункты могут отсутствовать.

Минимальный текст запроса, в котором есть только раздел описания запроса, может выглядеть, например, следующим образом:

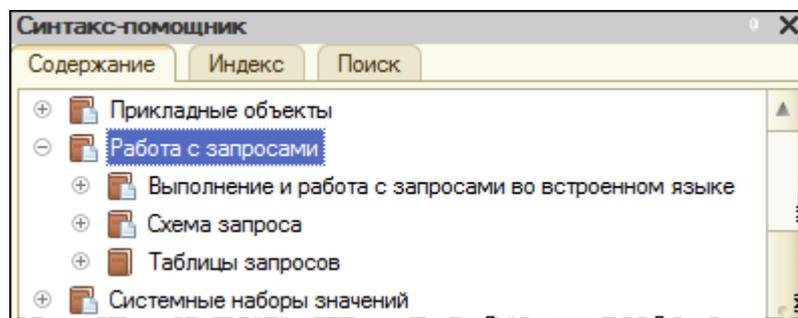
**ВЫБРАТЬ 1**

Этот запрос возвращает всегда одно значение – число 1.

Все разделы текста запроса будут подробно рассмотрены далее. Также во встроенной справке можно ознакомиться с полным описанием синтаксиса текста запроса. Для этого следует обратиться к разделу *Встроенный язык – Работа с запросами – Синтаксис текста запросов*.



Обратите внимание, что данный раздел присутствует только в справке, в *Синтакс-помощнике* такого раздела нет:



## Видеоурок. Консоль запросов

Данный урок записан в видеоформате. Обратитесь к файлам Query-2-Syntax-1 и Query-2-Syntax-1-add [на странице с материалами курса](#).

Изучив эти уроки, Вы узнаете:

- Как можно разрабатывать и отлаживать запросы в режиме «1С:Предприятие 8»
- Где скачать удобную консоль запросов с широкими функциональными возможностями.



## Видеоурок. Простой запрос

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-2 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Как выглядит минимальный запрос
- Как на практике пишется запрос в консоли
- Как выбрать все поля из справочника
- Как определить время выполнения запроса.



## Практическое задание №1. Вычисление выражений

В первом практическом задании необходимо получить значения нескольких выражений с помощью запросов.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Видеоурок. Конструктор запроса

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-3 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Для чего в платформе существует визуальный помощник – конструктор запроса
- Что конструктор запроса можно использовать и в конфигураторе, и в консоли запроса
- Как в конструкторе запроса просмотреть сформированный текст запроса и результат выполнения запроса
- По какому принципу построен интерфейс конструктора запроса.



## Видеоурок. Таблицы и поля

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-4 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Как в конструкторе запроса выбрать исходные таблицы и выходные поля
- Как конструктор повышает читаемость полученного текста запроса
- Как добавить в запрос собственное выходное поле.

## Псевдонимы полей

Выходному полю запроса можно присвоить псевдоним. После этого к этому полю можно обращаться, используя не имя, определенное в конфигураторе, а назначенный псевдоним. Псевдоним можно использовать в предложениях УПОРЯДОЧИТЬ ПО и ИТОГИ. Также важно, что далее при работе с результатом запроса во встроенном языке обращение к конкретному выходному полю будет вестись с использованием псевдонима. Если устанавливать псевдонимы со смыслом, то обращение к конкретному полю становится понятным и наглядным.

Перед псевдонимом может стоять ключевое слово КАК. Например:

```
ВЫБРАТЬ  
    Товары.Наименование КАК Заголовок  
ИЗ  
    Справочник.Товары КАК Товары
```

Ключевое слово КАК можно опустить, такой текст запроса также является правильным:

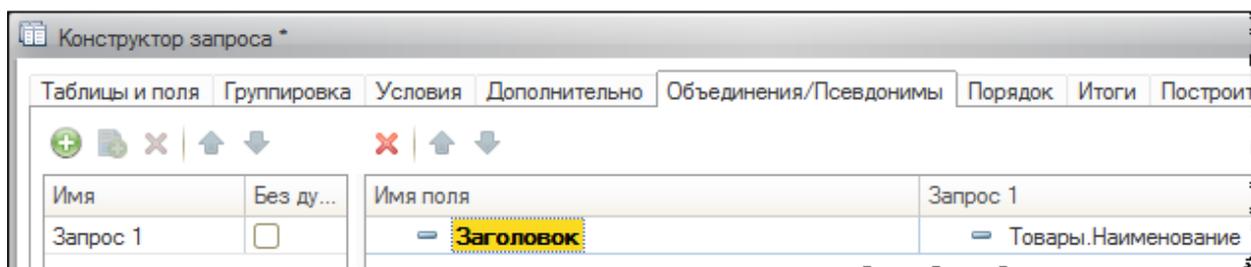
```
ВЫБРАТЬ  
    Товары.Наименование Заголовок  
ИЗ  
    Справочник.Товары Товары
```

Если текст запроса составляется при помощи конструктора запроса, то перед псевдонимами система всегда подставляет ключевое слово КАК.

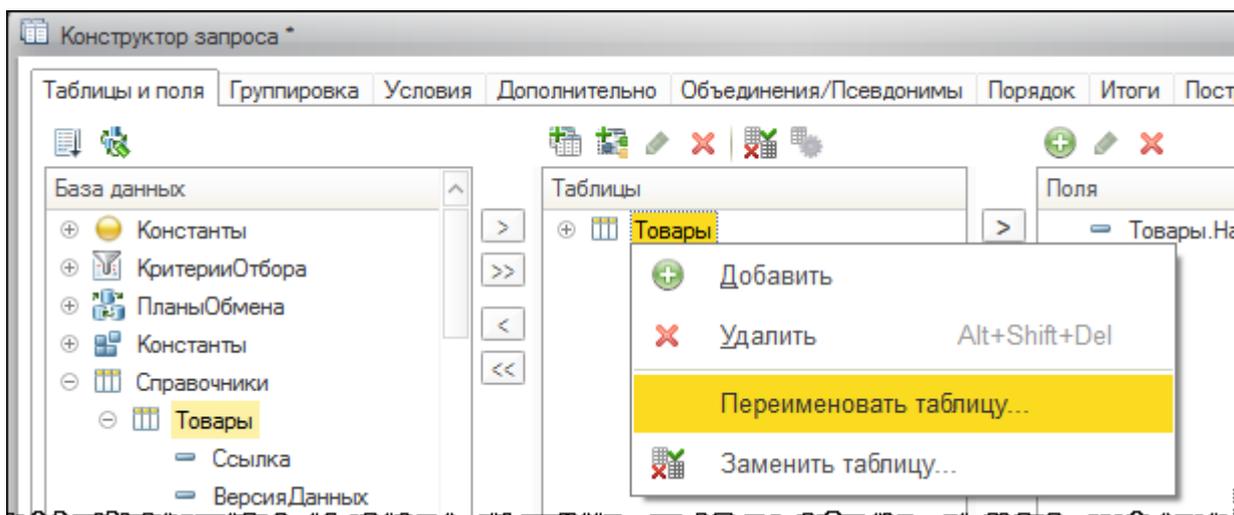
В отношении псевдонима существуют ограничения, действующие на любой идентификатор во встроенном языке. Также псевдонимы в запросе не могут повторяться.

Наличие или отсутствие псевдонимов не влияет на выборку самих данных из базы.

Чтобы при помощи конструктора запросов установить для поля псевдоним, следует перейти на закладку *Объединения/Псевдонимы* и отредактировать псевдоним поля в колонке *Имя поля*:



Также возможно присвоить псевдоним не только выходному полю, но и исходной таблице, из которой выбираются данные. Для этого в конструкторе запроса на закладке *Таблицы и поля* необходимо вызвать контекстное меню щелчком правой кнопки мыши и ввести псевдоним для таблицы:



## Комментарии в языке запросов

Комментарии в тексте запроса выделяются при помощи символов «//».

Часть строки от этих символов до конца строки считается комментарием и при выполнении запроса не учитывается. Важно, что при помощи конструктора запросов установить комментарии в тексте запроса нельзя, их можно внести только в сам текст запроса вручную. Также если текст запроса содержит комментарии, при открытии запроса в конструкторе, а затем при последующем сохранении комментарии будут удалены.

Если открыть текст запроса в конструкторе, а потом закрыть его, не сохраняя результат, то комментарии не пропадут.

## Группировка результатов запроса

Значения полей в таблицах могут повторяться. Например, один и тот же товар продавался в разных накладных. Поэтому может возникнуть вопрос, какое количество каждого товара и на какую сумму было реализовано.

Пусть исходные данные выглядят следующим образом:

Документ	Товар	Количество	Сумма (руб.)
Продажа 13 от 02.02.2014	Сапоги	2	7 000,00
Продажа 19 от 10.02.2014	Ботинки	1	2 000,00.
Продажа 28 от 17.02.2014	Сапоги	1	3 500,00

Тогда общие итоговые сведения о продажах за период будут выглядеть следующим образом:

Товар	Количество	Сумма (руб.)
Сапоги	3	10 500,00
Ботинки	1	2 000,00

Как видно из этой таблицы, в ней нет дублирующихся товаров, а значения по колонкам *Количество* и *Сумма* свернуты (просуммированы).

Для решения поставленной задачи применяется группировка в запросах. Смысл этой операции заключается в том, что вместо нескольких записей, имеющих одинаковое значение какого-либо поля, в результате запроса будет выведена только одна такая запись, а для других полей рассчитывается *агрегатная функция* (функция, применяемая сразу к нескольким записям набора данных, но возвращающая одно значение).

Так, для примера с продажей одинакового товара в разных накладных, записи будут группироваться по полю *Товар*, а количество товаров будет суммироваться.

Агрегатные функции указываются в списке полей выборки. В предложении СГРУППИРОВАТЬ ПО необходимо указать список полей, по которым следует произвести группировку. В запросе будут группироваться записи исходных таблиц, содержащие одинаковые значения указанных полей.

Важно, что в предложении СГРУППИРОВАТЬ ПО указываются имена полей, а не их псевдонимы.

Пример группировки данных в запросе:

ВЫБРАТЬ

РасходТовараТовары.Товар,  
СУММА(РасходТовараТовары.Количество) КАК Количество

ИЗ

Документ.РасходТовара.Товары КАК РасходТовараТовары

СГРУППИРОВАТЬ ПО

РасходТовараТовары.Товар

Пусть исходные данные в таблице *Документ.РасходТовара.Товары* имеют следующий вид:

Товар	Количество
-------	------------

Йогурт	12
Сметана	10
Молоко	5
Сметана	8
Молоко	4

Тогда после выполнения приведенного выше запроса будет получен следующий результат:

Товар	Количество
Йогурт	12
Сметана	18
Молоко	9

Важно понимать, что при использовании в запросе группировки все поля должны быть или полями группировки, или агрегатными функциями.

Допустимо использовать несколько раз агрегатные функции над одним и тем же полем:

ВЫБРАТЬ

```
РасходТовараТовары.Товар,  
МИНИМУМ(РасходТовараТовары.Количество) КАК Минимум,  
МАКСИМУМ(РасходТовараТовары.Количество) КАК Максимум
```

ИЗ

```
Документ.РасходТовара.Товары КАК РасходТовараТовары  
СГРУППИРОВАТЬ ПО  
РасходТовараТовары.Товар
```

При использовании агрегатных функций предложение СГРУППИРОВАТЬ ПО может быть опущено, тогда результат выполнения запроса будет представлять одну свернутую строку:

ВЫБРАТЬ

```
СУММА(РасходТовараТовары.Количество) КАК Количество
```

ИЗ

```
Документ.РасходТовара.Товары КАК РасходТовараТовары
```



## Практическое задание №2. Использование агрегатных функций

В ходе выполнения данного задания требуется определить максимальную цену закупки указанного пользователем товара.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Практическое задание №3. Определение количества записей в таблице базы данных

В данном задании необходимо с помощью запроса определить количество записей в любом регистре накопления.

Текст задания и его решение скачивайте со страницы с [материалами тренинга](#).

[Перейти к заданию](#)



## Практическое задание №4. Использование агрегатных функций для определения количества документов, делающих движения в регистр

В данном задании с помощью запроса необходимо определить количество документов, сделавших записи в любом регистре накопления.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#).

[Перейти к заданию](#)

## Упорядочивание результатов запроса

Упорядочивание результатов запроса – это правило хорошего тона при разработке. Без явного указания правила сортировки в запросе порядок записей неизвестен, его не знает заранее даже разработчик. Представление результатов запроса в таком неотсортированном виде очень неудобно для пользователя. А если алгоритм, реализованный разработчиком, будет ориентироваться на определенный порядок записей, то это потенциально может привести к некорректным результатам. Поэтому важно указывать, как должны быть отсортированы записи в запросе.

При помощи запроса выберем историю изменения цен на товары из регистра сведений «ЦеныТоваров»:

```
ЦеныТоваров.Период КАК Период,  
ЦеныТоваров.Товар,  
ЦеныТоваров.Цена
```

ИЗ

```
РегистрСведений.ЦеныТоваров КАК ЦеныТоваров
```

Период	Товар	Цена (руб.)
01.12.2010 0:00:00	Туфли	2 500,00
01.11.2011 0:00:00	Сапоги	3 050,00
19.06.2008 0:00:00	Босоножки	2 200,00

Для сортировки записей в тексте запроса используется предложение УПОРЯДОЧИТЬ ПО. В нем через запятую перечисляются условия, в соответствии с которыми необходимо упорядочить результат запроса.

Записи упорядочиваются сначала по первому условию, потом по второму и т.д. Например, необходимо выбрать историю изменения цен товаров, расположить в порядке возрастания даты изменения цены на товар:

```
ВЫБРАТЬ
```

```
ЦеныТоваров.Период КАК Период,  
ЦеныТоваров.Товар,  
ЦеныТоваров.Цена
```

ИЗ

```
РегистрСведений.ЦеныТоваров КАК ЦеныТоваров
```

```
УПОРЯДОЧИТЬ ПО
```

```
Период
```

В этом случае результат запроса будет следующий:

Период	Товар	Цена (руб.)
19.06.2008 0:00:00	Босоножки	2 200,00
01.12.2010 0:00:00	Туфли	2 500,00
01.11.2011 0:00:00	Сапоги	3 050,00

Упорядочивание может осуществляться в порядке возрастания или убывания значений. Порядок может задаваться для каждого поля независимо. Для указания порядка сортировки после имени поля необходимо указать ключевое слово ВОЗР (для сортировки по возрастанию) или УБЫВ (для сортировки по убыванию). Если тип упорядочивания не задан, то по умолчанию используется сортировка по возрастанию.

Поле, по которому производится упорядочивание, не обязательно должно входить в выходные поля.

ВЫБРАТЬ

ЦеныТоваров.Товар,  
ЦеныТоваров.Цена

ИЗ

РегистрСведений.ЦеныТоваров КАК ЦеныТоваров

УПОРЯДОЧИТЬ ПО

ЦеныТоваров.Период

При выполнении этого запроса записи будут расположены по возрастанию поля *Период*, хотя в выходные поля оно не входит.

Если производится упорядочивание по нескольким полям, то для каждого поля может быть указан свой порядок сортировки. Например:

ВЫБРАТЬ

ЦеныТоваров.Товар,  
ЦеныТоваров.Цена КАК Цена

ИЗ

РегистрСведений.ЦеныТоваров КАК ЦеныТоваров

УПОРЯДОЧИТЬ ПО

ЦеныТоваров.Период,  
Цена УБЫВ

Для справочников можно назначать упорядочивание по иерархии справочника. Для этого после имени поля указывается ключевое слово ИЕРАРХИЯ или ИЕРАРХИЯ УБЫВ:

ВЫБРАТЬ

Товары.Наименование КАК Наименование

ИЗ

Справочник.Товары КАК Товары

УПОРЯДОЧИТЬ ПО

Наименование ИЕРАРХИЯ

Наименование
<b>Обувь</b>
Босоножки
Ботинки
Валенки
<b>Продукты</b>
Йогурт
Колбаса
Масло

Жирным шрифтом выделены группы номенклатуры, за которыми следуют упорядоченные по наименованию элементы, находящиеся в этой группе.

## Автоупорядочивание

В предыдущих примерах была рассмотрена сортировка по полям с примитивными типами. Возникает вопрос – как упорядочить записи по ссылочному полю? Ведь поле *Ссылка* в базе данных представляет собой уникальный идентификатор, не несущий для пользователя смысловой нагрузки, поэтому непосредственная сортировка по значению ссылки не имеет практического смысла. Для справочников смысл заключается в поле *Код* или *Наименование*, для документов – в поле *Дата* и *Номер*.

Чтобы расположить записи таблиц в порядке, соответствующем смысловой нагрузке, используется автоупорядочивание результатов запросов. В тексте запросов этот режим устанавливается в специальной одноименной секции – АВТОУПОРЯДОЧИВАНИЕ.

С помощью этого режима можно вывести записи таблицы в наиболее привычном, ожидаемом пользователем порядке. Для этого нужно упорядочить записи таблицы по ссылочному полю с использованием конструкции АВТОУПОРЯДОЧИВАНИЕ:

ВЫБРАТЬ

Товары.Ссылка КАК Ссылка

ИЗ

Справочник.Товары КАК Товары

УПОРЯДОЧИТЬ ПО

Ссылка

АВТОУПОРЯДОЧИВАНИЕ

Ссылка
<b>Обувь</b>
<b>Продукты</b>
Босоножки
Ботинки
Валенки
Йогурт
Колбаса
Масло

При автоупорядочивании записи расположены в порядке возрастания поля *Наименование*, заданного для справочника в качестве основного представления. Жирным шрифтом в таблице выделены группы номенклатуры.

Автоупорядочивание работает по следующим принципам:

1. Если в запросе было указано предложение УПОРЯДОЧИТЬ ПО, то каждая ссылка на таблицу, находящаяся в этом предложении, будет заменена полями, по которым по умолчанию сортируется таблица (для справочников это код или наименование, для документов – дата документа). Если поле для упорядочивания ссылается на иерархический справочник, то будет применена иерархическая сортировка по этому справочнику
2. Если в запросе отсутствует предложение УПОРЯДОЧИТЬ ПО, но есть предложение ИТОГИ, тогда результат запроса будет упорядочен по полям, присутствующим в предложении ИТОГИ после ключевого слова ПО, в той же последовательности. Если среди полей из предложения ИТОГИ есть поля ссылочного типа, то результат также будет упорядочен по полям, по которым по умолчанию сортируется таблица этого типа.
3. Если в запросе отсутствуют предложения УПОРЯДОЧИТЬ ПО и ИТОГИ, но есть предложение СГРУППИРОВАТЬ ПО, тогда результат запроса будет упорядочен по полям, присутствующим в предложении, в той же последовательности. Если среди группировочных полей есть поля ссылочного типа, то результат также будет упорядочен по полям, по которым по умолчанию сортируется таблица этого типа.
4. Если в запросе отсутствуют предложения УПОРЯДОЧИТЬ ПО, ИТОГИ и СГРУППИРОВАТЬ ПО, результат будет упорядочен по полям сортировки по умолчанию для таблиц, из которых выбираются данные, в порядке их появления в запросе.



## Видеоурок. Текстовое представление ссылочного поля

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-5 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Что представляет собой значение поля *Ссылка*
- Как можно улучшить производительность системы при выводе ссылочного поля в отчет
- Особенность функции языка запроса ПРЕДСТАВЛЕНИЕ()
- Как можно переопределить представление в платформе 8.3.



## Видеоурок. Функция ПРЕДСТАВЛЕНИЕ()

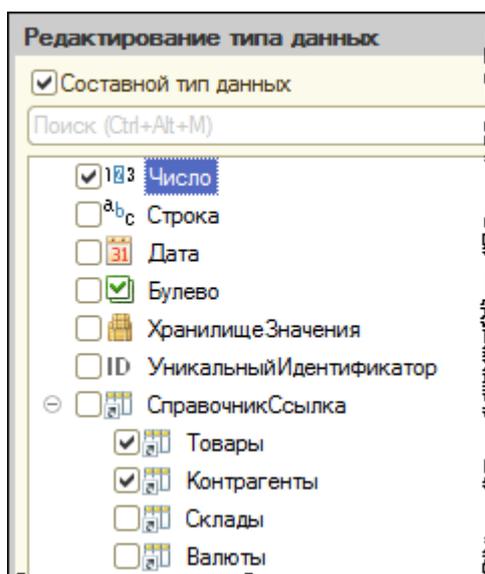
Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-6 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Для каких типов данных доступна функция языка запросов ПРЕДСТАВЛЕНИЕ()
- В чем отличие функций языка запросов ПРЕДСТАВЛЕНИЕ() и ПРЕДСТАВЛЕНИЕССЫЛКИ().

## Отличие поля Представление от функции ПРЕДСТАВЛЕНИЕ()

Пусть у справочника «Товары» существует реквизит *Реквизит1*, который имеет составной тип. В составной тип входят ссылочный и примитивный (например, число) типы:



Выполним для этого справочника следующий запрос:

ВЫБРАТЬ

```
Товары.Ссылка,
Товары.Реквизит1,
Товары.Реквизит1.Представление,
ПРЕДСТАВЛЕНИЕ(Товары.Реквизит1) КАК Реквизит1Представление1
```

ИЗ

```
Справочник.Товары КАК Товары
```

Если реквизит хранит значение примитивного типа (например, число 555), то для него представление, полученное при помощи виртуального поля *Представление*, не будет получено. Если для такого поля применить функцию *ПРЕДСТАВЛЕНИЕ()*, то оно будет получено вне зависимости от типа значения, хранящегося в поле.

Ссылка	Реквизит1	Реквизит1Представление	Реквизит1Представление1
Босоножки	555	Null	555
Ботинки	Попов Б.В. ИЧП	Попов Б.В. ИЧП	Попов Б.В. ИЧП

При использовании функции *ПРЕДСТАВЛЕНИЕ()* значение представления может получаться одним или несколькими дополнительными запросами к базе данных.

В основном запросе из базы данных будут извлекаться только ссылки. Это позволяет оптимальнее получать представление для реквизитов, в состав которых входят ссылки на большое количество таблиц (например, тип *СправочникСсылка* или *ЛюбаяСсылка*). Лучшая производительность достигается за счет того, что в запрос не будут помещаться

неявные соединения для получения представления реквизита. Функция ПРЕДСТАВЛЕНИЕ() также может использоваться при выполнении запроса через СОМ-соединение.

## Ключевое слово ПЕРВЫЕ

Если в результате запроса необходимо вывести только ограниченное число записей, то в секции описания запроса следует указать ключевое слово ПЕРВЫЕ, после чего ввести число записей, которое следует вывести в результат запроса. Например, такой запрос вернет только 5 первых записей из таблицы *Товары*:

```
ВЫБРАТЬ ПЕРВЫЕ 5  
    Товары.Наименование  
ИЗ  
    Справочник.Товары КАК Товары
```

Ключевое слово ПЕРВЫЕ эффективно использовать совместно с сортировкой по убыванию какого-либо поля. Например, отобрать 3 самых дорогих товара, проданных покупателям:

```
ВЫБРАТЬ ПЕРВЫЕ 3  
    РасходТовараТовары.Товар,  
    РасходТовараТовары.Цена КАК Цена  
ИЗ  
    Документ.РасходТовара.Товары КАК РасходТовараТовары  
УПОРЯДОЧИТЬ ПО  
    Цена УБЫВ
```

При выполнении этого запроса будет получен следующий результат:

Товар	Цена (руб.)
Veko67NE	15 000,00
Veko345MO	10 000,00.
Sony K3456P	4 500,00



## Практическое задание №5. **Получение определенного количества записей в запросе**

В данном практическом задании необходимо в результате запроса получить данные о трех клиентах, совершивших покупки на наибольшую сумму.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)

## Ключевое слово РАЗЛИЧНЫЕ

Часто бывает, нужно исключить из отчета одинаковые строки. Например, чтобы узнать, каким вообще покупателям отгружался товар, можно использовать запрос:

**ВЫБРАТЬ**

РасходТовара.Покупатель

**ИЗ**

Документ.РасходТовара КАК РасходТовара

Покупатель
Попов Б.В. ИЧП
Попов Б.В. ИЧП
Шлюзовая ООО
Магазин «Мясная лавка»
Магазин «Продукты»
Шлюзовая ООО
Шлюзовая ООО

В результате запроса представлено несколько повторяющихся строк, что снижает его наглядность. Чтобы избежать повторов, в описании запроса следует указать ключевое слово РАЗЛИЧНЫЕ:

**ВЫБРАТЬ** РАЗЛИЧНЫЕ

РасходТовара.Покупатель

**ИЗ**

Документ.РасходТовара КАК РасходТовар

Покупатель
Попов Б.В. ИЧП
Шлюзовая ООО
Магазин «Мясная лавка»
Магазин «Продукты»

Если в запросе несколько выходных полей, то при использовании ключевого слова РАЗЛИЧНЫЕ из результата запроса будут исключены повторяющиеся строки. Т.е. в

результате запроса будут только строки с различными значениями выходных полей, полностью повторяющиеся строки будут отброшены. Рассмотрим пример:

ВЫБРАТЬ

```
РасходТовараТовары.Товар КАК Товар,  
РасходТовараТовары.Цена
```

ИЗ

```
Документ.РасходТовара.Товары КАК РасходТовараТовары
```

Этот запрос возвращает следующий результат:

Товар	Цена (руб.)
Молоко	30,00
Колбаса	250,00
Колбаса	250,00
Йогурт	30,00
Йогурт	40,00
Молоко	30,00

Если в текст запроса добавить ключевое слово РАЗЛИЧНЫЕ, то результат изменится:

Товар	Цена (руб.)
Молоко	30,00
Колбаса	250,00
Йогурт	30,00
Йогурт	40,00



## Видеоурок. Условия в тексте запроса

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-7 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Как ограничить выборку данных
- Как интерактивно настроить условие в запросе при помощи конструктора
- Какие особенности существуют в языке запросов при сравнении строк.



## Практическое задание №6. Условия в запросе

Во втором практическом задании из документов «Поступление товаров» необходимо будет получить список товаров, исходя из их закупочной стоимости.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)

## Оператор ПОДОБНО

Оператор ПОДОБНО позволяет сравнить значение выражения, указанного слева от него, со строкой шаблона, указанной справа. Значение выражения должно иметь тип *Строка*. Если значение выражения удовлетворяет шаблону, результатом оператора будет *Истина*, иначе – *Ложь*. Следовательно, можно воспользоваться этим оператором в предложении ГДЕ текста запроса.

Следующие символы в строке шаблона являются служебными и имеют смысл, отличный от символа строки:

- «%» (процент): последовательность, содержащая любое количество произвольных символов
- «\_» (подчеркивание): один произвольный символ
- «[...]» (в квадратных скобках один или несколько символов): любой одиночный символ из перечисленных внутри квадратных скобок. В перечислении могут встречаться диапазоны, например, a-z, означающие произвольный символ, входящий в диапазон, включая концы диапазона
- «[^...]» (в квадратных скобках значок отрицания, за которым следует один или несколько символов): любой одиночный символ, кроме тех, которые перечислены следом за значком отрицания.

Любой другой символ означает сам себя и не несет никакой дополнительной нагрузки.

Если в качестве самого себя необходимо записать один из перечисленных символов, то ему должен предшествовать спецсимвол. Сам спецсимвол (любой подходящий символ) определяется в этом же операторе после ключевого слова СПЕЦСИМВОЛ.

При помощи оператора ПОДОБНО можно проверять соответствие строки определенной маске. Например, необходимо выбрать всех контрагентов, у которых в поле *Город* содержится подстрока «моск», причем с любой позиции:

ВЫБРАТЬ

Контрагенты.Наименование,  
Контрагенты.Город

ИЗ

Справочник.Контрагенты КАК Контрагенты

ГДЕ

Контрагенты.Город ПОДОБНО "%МОСК%"

Чтобы найти контрагентов, у которых поле *ДополнительнаяИнформация* содержит символ подчеркивания («\_»), необходимо составить следующий запрос:

ВЫБРАТЬ

Контрагенты.Наименование,  
Контрагенты.Город

ИЗ

Справочник.Контрагенты КАК Контрагенты

ГДЕ

Контрагенты.ДополнительнаяИнформация ПОДОБНО "%\\_%" СПЕЦСИМВОЛ "\\_"

Существуют особенности при использовании оператора ПОДОБНО в клиент-серверном варианте в зависимости от используемой СУБД.

При использовании СУБД IBM DB2 правым операндом оператора ПОДОБНО может быть только литерал (параметр) или выражение над литералами. Шаблонными символами являются только «\_» – любой символ и «%» – последовательность любых символов.

При использовании СУБД PostgreSQL и Oracle Database в запросах, содержащих ПОДОБНО, квадратные скобки в шаблоне воспринимаются как специальные символы только в том случае, если они используются в текстовом литерале, а не в выражении.

Например, в этом случае квадратные скобки обозначают множество символов:

```
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Товары КАК Товары
ГДЕ
    Товары.Артикул ПОДОБНО "123[АБВ]%"
```

А в этом примере квадратные скобки, которые могут находиться в параметре *Шаблон*, не воспринимаются как специальные символы:

```
ВЫБРАТЬ
    Товары.Наименование
ИЗ
    Справочник.Товары КАК Товары
ГДЕ
    Товары.Артикул ПОДОБНО &Шаблон + "%"
```

Необходимо учитывать эти особенности, поскольку при разработке на файловой базе запрос будет обрабатываться корректно, а на рабочей базе в клиент-серверном варианте может возникнуть неожиданное поведение.



## Практическое задание №7. **Использование спецсимвола в запросе**

В ходе выполнения практического задания нужно получить список товаров, в наименовании которых встречается специальный символ.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Практическое задание №8. Использование оператора ПОДОБНО

В данном практическом задании необходимо запросом получить список клиентов, у которых номер телефона не соответствует стандарту.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## **Видеоурок. Использование параметров**

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-8 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Зачем в запросе нужны параметры
- Как указать в запросе параметр через амперсанд (&), не переключаясь на английскую раскладку клавиатуры
- Как в консоли запросов автоматически получить все параметры, использованные в тексте запроса.



## Практическое задание №9. **Использование параметров и ключевого слова РАЗЛИЧНЫЕ в запросе**

В третьем практическом задании из документов «Реализация товаров» необходимо выбрать всех клиентов, приобретавших заданный пользователем товар.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Практическое задание №10. Работа со строками

В данном задании с помощью запроса необходимо получить список контрагентов из указанного пользователем города.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Практическое задание №11. Работа с подчиненными справочниками

В данном практическом задании необходимо получить с помощью запроса всех контрагентов, имеющих расчетные счета в указанном пользователем банке.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## **Видеоурок. Замена таблицы в запросе при помощи конструктора**

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-9 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Как заменить в тексте запроса ошибочно выбранную таблицу, не переписывая весь запрос.



## **Видеоурок. Использование вложенных таблиц в запросе**

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-10 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Как в запросе указать в качестве выходного поля вложенную таблицу
- В чем отличие обычного выходного поля от вложенной таблицы
- Как доработать используемую консоль запросов, чтобы можно было просмотреть содержимое вложенной таблицы в результате запроса.

## Табличные части справочников и документов в запросах

В языке запросов можно обращаться к табличным частям справочников, документов и других объектов метаданных. В базе данных табличные части хранятся как отдельные таблицы.

При добавлении в конфигураторе в справочник табличной части в информационной базе создается таблица со стандартными реквизитами *Ссылка* и *НомерСтроки* и реквизитами табличной части, заданными в конфигураторе. Таблица, содержащая табличную часть, связана по полю *Ссылка* с основной таблицей. Благодаря этому можно получить информацию из табличной части, относящуюся к конкретному элементу справочника. В информационной базе создается столько подчиненных таблиц, сколько табличных частей задано у справочника.

### Пример 1.

#### Вывести документ продажи, проданный товар, количество, сумму

Данные будем выбирать непосредственно из табличной части *Товары*:

ВЫБРАТЬ

```
РасходТовараТовары.Ссылка,  
РасходТовараТовары.Товар,  
РасходТовараТовары.Количество,  
РасходТовараТовары.Сумма
```

ИЗ

```
Документ.РасходТовара.Товары КАК РасходТовараТовары
```

Ссылка	Товар	Количество	Сумма (руб.)
Продажа 000000016 от 10.09.2012 14:21:55	Йогурт	300	10 500,00
Продажа 000000017 от 12.09.2012 15:30:45	Йогурт	300	12 000,00
Продажа 000000010 от 18.08.2012 12:25:55	Сапоги	2	6 600,00

Следует обратить внимание, что при указании вложенной таблицы в списке источников допускается обращение как к полям самой вложенной таблицы, так и к полям таблицы верхнего уровня через поле *Ссылка*.

### Пример 2.

#### Вывести покупателя, проданный товар, количество, сумму

ВЫБРАТЬ

```
РасходТовараТовары.Ссылка.Покупатель,  
РасходТовараТовары.Товар,
```

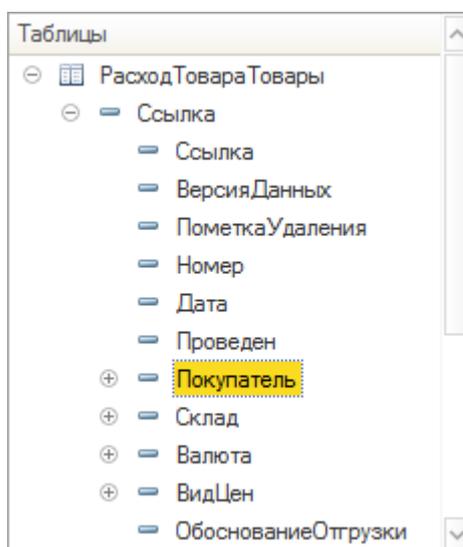
РасходТовараТовары.Количество,  
РасходТовараТовары.Сумма

из

Документ.РасходТовара.Товары КАК РасходТовараТовары

Покупатель	Товар	Количество	Сумма (руб.)
Магазин «Продукты»	Йогурт	300	10 500,00
Попов Б.В. ИЧП	Йогурт	300	12 000,00
Попов Б.В. ИЧП	Сапоги	2	6 600,00

Показанный прием способствует большей понятности запроса – при работе с табличной частью поля основной таблицы доступны как вложенные поля у поля *Ссылка*.



Система при таком обращении неявно добавит в запрос соединение с основной таблицей.

## Функция ЗНАЧЕНИЕ()

Обращение в запросах к predetermined данным конфигурации и значениям системных перечислений осуществляется с помощью функции ЗНАЧЕНИЕ().

В тексте запроса могут указываться следующие predetermined данные конфигурации:

- Значения перечислений
- Предetermined данные справочников, планов видов характеристик, планов счетов, планов видов расчетов
- Пустые ссылки
- Значения точек маршрута бизнес-процессов.

Также текст запроса может содержать значения системных перечислений, которые могут быть присвоены полям в таблицах базы данных: *ВидДвиженияНакопления*, *ВидСчета* и *ВидДвиженияБухгалтерии*.

В качестве имени объекта метаданных указывается имя объекта метаданных, как оно задано в конфигураторе. Например, обращение в тексте запроса к значению перечисления:

```
ВЫБРАТЬ  
    Товары.Наименование  
ИЗ  
    Справочник.Товары КАК Товары  
ГДЕ  
    Товары.Вид = ЗНАЧЕНИЕ (Перечисление.ВидыТоваров.Товар)
```

Обращение к predetermined элементу справочника (вид цены Закупочная создан как predetermined в конфигураторе) выглядит следующим образом:

```
ВЫБРАТЬ  
    Контрагенты.Наименование  
ИЗ  
    Справочник.Контрагенты КАК Контрагенты  
ГДЕ  
    Контрагенты.ВидЦен = ЗНАЧЕНИЕ (Справочник.ВидыЦен.Закупочная)
```

Обращение к пустой ссылке в языке запросов:

```
ВЫБРАТЬ  
    Контрагенты.Наименование  
ИЗ  
    Справочник.Контрагенты КАК Контрагенты  
ГДЕ  
    Контрагенты.ВидЦен = ЗНАЧЕНИЕ (Справочник.ВидыЦен.ПустаяСсылка)
```

Также возможен второй вариант обращения к пустой ссылке:

```
ГДЕ  
    Контрагенты.ВидЦен = ЗНАЧЕНИЕ (Справочник.ВидыЦен.)
```

Пример обращения к значению системного перечисления *ВидДвиженияНакопления*:

```
ВЫБРАТЬ  
    ТоварныеЗапасы.Регистратор,  
    ТоварныеЗапасы.Товар,  
    ТоварныеЗапасы.Количество  
ИЗ  
    РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы  
ГДЕ  
    ТоварныеЗапасы.ВидДвижения = ЗНАЧЕНИЕ (ВидДвиженияНакопления.Приход)
```



## Практическое задание №12. **Использование в запросах предопределенных данных конфигурации**

В данном задании необходимо с помощью запроса выбрать из справочника все позиции, являющиеся услугами.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Видеоурок. Функции для работы с датами

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-11 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Какие функции для работы с датами существуют в языке запросов
- Как в запросе выделить составную часть (год, месяц, день и т.д.) из даты
- Как в запросе собрать дату из составных частей (год, месяц, день и т.д.)
- Какую максимальную дату можно указать в тексте запроса.



## Практическое задание №13. Использование ключевого слова **РАЗЛИЧНЫЕ**

В этом задании необходимо с помощью запроса получить список контрагентов, поставивших товар, и даты поставок.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Видеоурок. Выбор данных из иерархических справочников

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-12 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Как отсортировать элементы справочника согласно их расположению в иерархии справочника
- Как интерактивно в конструкторе установить иерархическое упорядочивание.



## Видеоурок. Работа с иерархическими справочниками

Данный урок записан в видеоформате. Обратитесь к файлам Query-2-Syntax-13 и Query-2-Syntax-14 [на странице с материалами курса](#).

Изучив эти уроки, Вы узнаете:

- Как в результате запроса к справочнику может быть получено значение типа NULL
- Как отобразить записи из конкретной группы
- Как отобразить записи, находящиеся в корне
- Как отобразить записи, находящиеся в конкретной группе и во всех подчиненных ей группах.



## **Видеоурок. Использование конструкции ИМЕЮЩИЕ**

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-15 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Как наложить фильтр на поле, к которому применена агрегатная функция
- В чем отличие конструкции ИМЕЮЩИЕ от ГДЕ
- Как визуально в конструкторе запроса будут отличаться условия, наложенные при помощи ГДЕ и ИМЕЮЩИЕ.



## Практическое задание №14. Применение условий к агрегатным функциям

В этом практическом задании необходимо выбрать товары, исходя из их суммарной стоимости закупа.

Текст задания и его решение скачивайте со [страницы с материалами тренинга:](#)

[Перейти к заданию](#)



## Видеоурок. Оператор ВЫБОР

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-16 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Как в запросе выбрать одно из возможных значений в зависимости от условий
- Как выглядит недокументированный синтаксис оператора ВЫБОР.



## Практическое задание №15. Использование оператора **ВЫБОР**

В данном практическом задании необходимо запросом получить список документов «Поступление товаров», попадающих в заявленный интервал.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#):

[Перейти к заданию](#)



## Видеоурок. Строки неограниченной длины

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-17 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Какие ограничения существуют в языке запросов при работе со строками неограниченной длины
- Как конвертировать строку неограниченной длины в обычную строку
- Можно ли в языке запросов привести число к строке или наоборот.

## Типы данных в запросах

В языке запросов существует несколько конструкций для работы с типами данных.

**ТИП (<Имя типа>)**

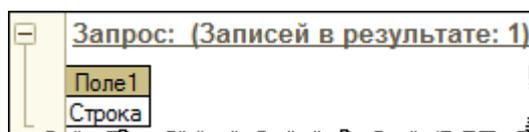
<Имя типа> – имя примитивного типа либо имя таблицы, тип ссылки которой нужно получить. Результатом данной конструкции будет значение типа *Тип*.

**Пример 1.**

**Получить в запросе тип *Строка***

**ВЫБРАТЬ ТИП (Строка)**

Результат такого запроса будет выглядеть следующим образом:

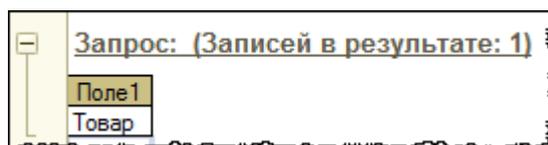


**Пример 2.**

**Получить в запросе тип ссылка на справочник «Товары»**

**ВЫБРАТЬ ТИП (Справочник.Товары)**

Результат такого запроса будет выглядеть следующим образом:



Значения типа *Тип* в языке запросов можно использовать в операциях сравнения, упорядочивания, группировки.

Функция ТИПЗНАЧЕНИЯ() предназначена для определения типа значения выражения любого типа в запросе. Возвращаемое значение – значение типа *Тип*. Например:

```
ВЫБРАТЬ  
    ТИПЗНАЧЕНИЯ (Взаиморасчеты.Регистратор) КАК Документ  
ИЗ  
    РегистрНакопления.Взаиморасчеты КАК Взаиморасчеты
```

Результат такого запроса будет выглядеть следующим образом:

Запрос: РегистрНакопления.Взаиморасчеты (Записей в результате: 98)

Документ
Поступление денег
Оплата
Продажа
Оплата

### Пример 3.

Необходимо выбрать из регистра накопления «ТоварныеЗапасы» записи, в которых поле *Регистратор* имеет тип ссылки на документ *РасходТовара*

```
ВЫБРАТЬ  
    ТИПЗНАЧЕНИЯ (ТоварныеЗапасы.Регистратор)  
ИЗ  
    РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы  
ГДЕ  
    ТИПЗНАЧЕНИЯ (ТоварныеЗапасы.Регистратор) = ТИП (Документ.РасходТовара)
```

Возможна также передача значения типа *Тип* как параметра запроса:

```
ВЫБРАТЬ  
    ТИПЗНАЧЕНИЯ (ТоварныеЗапасы.Регистратор)  
ИЗ  
    РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы  
ГДЕ  
    ТИПЗНАЧЕНИЯ (ТоварныеЗапасы.Регистратор) = &Тип
```

Этот же результат можно получить при помощи логической операции ССЫЛКА.

Этот оператор позволяет проверить, является ли значение выражения, указанного слева от него, ссылкой на таблицу, указанную справа. Если да, результатом оператора будет *Истина*, иначе – *Ложь*. Например:

```
ВЫБРАТЬ  
    ТИПЗНАЧЕНИЯ (ТоварныеЗапасы.Регистратор)
```

ИЗ

РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы

ГДЕ

ТоварныеЗапасы.Регистратор ССЫЛКА Документ.РасходТовара

**Пример 4.****Необходимо привести строку неограниченной длины к строке ограниченной длины**

Функцию языка запросов ВЫРАЗИТЬ() можно использовать для преобразования строки неограниченной длины к строке определенной ограниченной длины. Например, во многих документах в типовых конфигурациях присутствует реквизит *Комментарий неограниченной длины*. Такой реквизит нельзя использовать в запросах в качестве группового поля. Но если ограничить длину такой строки, то по ней можно будет выполнять группировку. Чтобы определить количество одинаковых комментариев в документах реализации, можно использовать следующий запрос:

ВЫБРАТЬ

КОЛИЧЕСТВО (РАЗЛИЧНЫЕ РеализацияТоваровУслуг.Ссылка) КАК Ссылка,

ВЫРАЗИТЬ (РеализацияТоваровУслуг.Комментарий КАК СТРОКА (200)) КАК Комментарий

ИЗ

Документ.РеализацияТоваровУслуг КАК РеализацияТоваровУслуг

СГРУППИРОВАТЬ ПО

ВЫРАЗИТЬ (РеализацияТоваровУслуг.Комментарий КАК СТРОКА (200))

**Пример 5.****Необходимо в запросе изменить точность числового значения**

Если в запросе необходимо получить поле числового типа, которое имеет определенную длину и точность, то можно воспользоваться функцией языка запроса ВЫРАЗИТЬ(). Например, если при расчете цены по сумме и количеству нет надобности в большой точности результата, достаточно двух знаков после запятой.

В запросе это можно реализовать следующим образом:

ВЫБРАТЬ

ПродажиОбороты.Покупатель,

ПродажиОбороты.Товар,

ПродажиОбороты.КоличествоОборот КАК Количество,

ПродажиОбороты.СуммаОборот КАК Сумма,

ВЫБОР

КОГДА ПродажиОбороты.КоличествоОборот = 0

ТОГДА 0

ИНАЧЕ ВЫРАЗИТЬ (ПродажиОбороты.СуммаОборот /

ПродажиОбороты.КоличествоОборот КАК ЧИСЛО (15, 2))

КОНЕЦ КАК Цена

ИЗ

РегистрНакопления.Продажи.Обороты КАК ПродажиОбороты

Обратите внимание, что в этом случае при использовании функции ВЫРАЗИТЬ() происходит округление числа. Например, следующий запрос вернет результат 16:

ВЫБРАТЬ

ВЫРАЗИТЬ (15.87 КАК ЧИСЛО (15, 0))

### Пример 6.

#### Необходимо в запросе привести значение составного типа к определенному заданному типу

Операция приведения типов ВЫРАЗИТЬ() позволяет привести значение составного типа к одному из составляющих это значение типов. Например, если известно, что поле составного типа является ссылкой на конкретную таблицу, то рекомендуется использовать функцию ВЫРАЗИТЬ() для ограничения количества таблиц в запросе.

ВЫБРАТЬ

ВЫРАЗИТЬ (ТоварныеЗапасы.Регистратор КАК Документ.РасходТовара) .Номер КАК  
Номер,

ВЫРАЗИТЬ (ТоварныеЗапасы.Регистратор КАК Документ.РасходТовара) .Дата КАК  
Дата  
ИЗ

РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы

ГДЕ

ТоварныеЗапасы.Регистратор ССЫЛКА Документ.РасходТовара

Это позволит значительно ускорить работу запроса, ограничив количество неявных соединений, которые будут добавлены в запрос при разыменовании полей. А при использовании, например, MS SQL Server существует ограничение на количество используемых таблиц в одном запросе – не более 256. Таким образом, использование в запросе поля составного типа может привести теоретически к неработоспособности запроса на определенной СУБД.

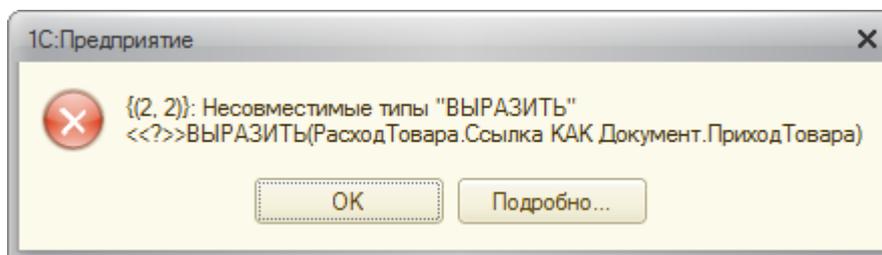
Не все типы данных можно привести друг к другу при помощи функции ВЫРАЗИТЬ(). Если попытаться преобразовать, например, ссылку на документ *РасходТовара* в ссылку на документ *ПриходТовара*, то будет выведено сообщение об ошибке.

ВЫБРАТЬ

ВЫРАЗИТЬ (РасходТовара.Ссылка КАК Документ.ПриходТовара)

ИЗ

Документ.РасходТовара КАК РасходТовара





## Видеоурок. Особенность предложения ИЗ

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-18 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Можно ли опустить ключевое слово ИЗ в тексте запроса.



## Видеоурок. Особенность СГРУППИРОВАТЬ ПО

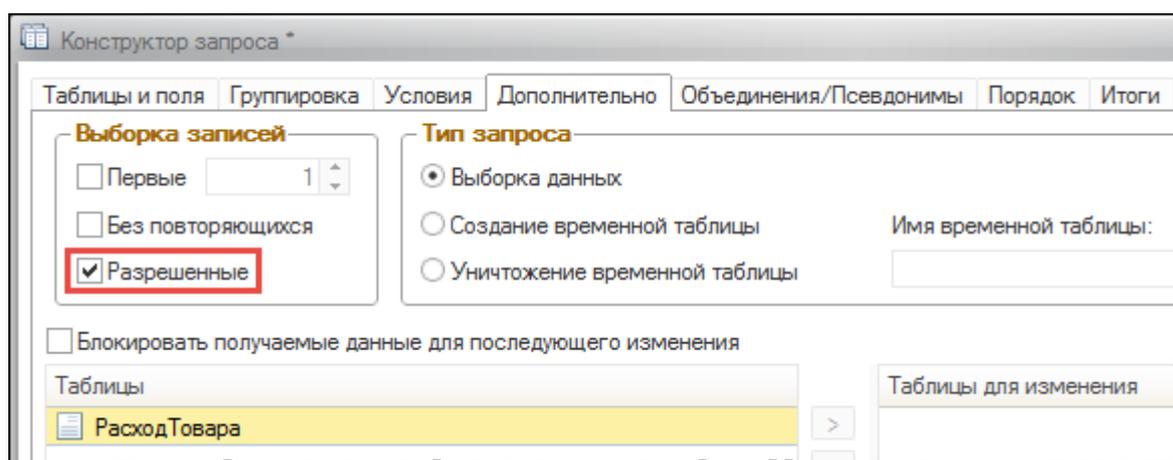
Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-19 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Какие различные результаты может вернуть запрос, если источник данных окажется пустым.

## Ключевое слово РАЗРЕШЕННЫЕ

Ключевое слово РАЗРЕШЕННЫЕ означает, что запрос выберет только те записи, на которые у текущего пользователя есть права. В конструкторе запроса для добавления к тексту запроса этого ключевого слова необходимо установить флаг *Разрешенные* на закладке *Дополнительно*:



ВЫБРАТЬ РАЗРЕШЕННЫЕ

РасходТовара.Ссылка,  
РасходТовара.Покупатель,  
РасходТовара.Склад

ИЗ

Документ.РасходТовара КАК РасходТовара

Если данный критерий не указать, то запрос отработает с ошибкой, выбрав записи, на которые у пользователя нет прав. Данное ключевое слово может присутствовать только в предложении ВЫБРАТЬ верхнего уровня и распространяется на весь запрос, включая вложенные запросы.

Следует учитывать, что использование ключевого слова РАЗРЕШЕННЫЕ оказывает влияние на работу только в том случае, если на таблицы наложены ограничения доступа к данным (RLS).

Механизм ограничений доступа к данным позволяет управлять правами доступа не только к конкретному объекту метаданных, но и к конкретному объекту базы данных.

Для чтения различных полей объекта могут быть заданы различные условия. Для условия можно указать имя конкретного поля или *Прочие поля*. В первом случае условие будет накладываться только в том случае, если в запросе присутствует поле с ограничением, а во втором – ограничение будет накладываться для всех полей объекта, кроме полей, для которых ограничения заданы явно.

Ограничения доступа к данным:	
Поля	Ограничение доступа
<Прочие поля>	РасходТовара ГДЕ РасходТовара.Проведен
Ссылка, ВерсияДанных, ПометкаУдаления, Номер, Дата	

Если ограничение доступа установлено на конкретное поле, такое поле будет получено из базы, если ограничение выполняется.

Если ограничение доступа установлено на Прочие поля, данные будут получены, если ограничение выполняется для всех полей объекта, участвующих в запросе.

Язык ограничения доступа к данным похож на язык запросов, но имеет следующие особенности. В запросе ограничения доступа к данным всегда присутствует одна таблица в качестве источника данных – это таблица объекта, на который накладывается ограничение. В языке ограничения доступа к данным используются только секции ИЗ и ГДЕ языка запросов. В секции ГДЕ допустимо указывать в качестве параметров функциональные опции и параметры сеанса. Также в языке ограничения доступа к данным для упрощения можно использовать специально разработанные шаблоны.

Обратите внимание, что если на таблицу отсутствует право ЧТЕНИЕ, то запрос с использованием такой таблицы, будет выполнен с ошибкой вне зависимости от использования в запросе ключевого слова РАЗРЕШЕННЫЕ.



## Видеоурок. Особенности конструктора запросов

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-20 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Как конструктор запроса интерпретирует звездочку (\*) в списке полей выборки
- Какая неприятная ситуация может возникнуть в случае использования комментариев в тексте запроса.



## Видеоурок. Управляемый конструктор запроса

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-21 [на странице с материалами курса](#)

Изучив этот урок, Вы узнаете:

- Можно ли открыть конструктор запроса в тонком клиенте или веб-клиенте
- Какие визуальные новшества внесли разработчики в управляемый конструктор запроса
- Как реализовать свою собственную простейшую консоль запроса для управляемого предложения.



## Видеоурок. Объектная модель запроса

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-22 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Как программно скомпоновать текст запроса, пользуясь объектами встроенного языка
- Как при помощи новых возможностей платформы 8.3.5 можно избежать составления текста запроса из отдельных фрагментов.



## Видеоурок. Отображение плана запроса в консоли запросов

Данный урок записан в видеоформате. Обратитесь к файлу Query-2-Syntax-23 [на странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Как получить план выполнения запроса при помощи только консоли запроса

- Как определить текст SQL-запроса, который будет реально исполняться в СУБД
- Какой механизм использует консоль запросов для отображения плана выполнения запроса.



## Практическое задание №16. Разыменование полей

В данном практическом задании необходимо получить запросом артикулы всех товаров, встречающихся в табличной части документа.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Практическое задание №17. Получение запросом даты документа- регистратора из регистра

В ходе выполнения данного практического задания необходимо при помощи запроса получить даты всех документов, осуществляющих движения в регистр накопления.

Текст задания и его решение скачивайте [со страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Практическое задание №18. **Переопределение основного представления справочника в платформе 8.3. Виртуальное поле Представление в запросе**

В этом практическом задании нужно будет разработать запрос для справочника «Номенклатура».

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#)

[Перейти к заданию](#)



## Вопросы и ответы в Мастер-группе

Если у Вас возникли вопросы по материалам текущего модуля курса, Вы можете [здать их на странице сайта](#):

[Перейти к вопросам](#)

На этой же странице Вы можете просмотреть ответы на вопросы других участников курса и почерпнуть новые знания. Данная возможность доступна только участникам платной версии тренинга.

## Модуль 3. Работа с итогами в тексте запроса

### Предназначение итогов

Часто требуется не только вывести определенные данные в результат запроса, но и рассчитать по ним некоторые обобщенные данные – итоги. Похожая задача решалась при помощи группировки в запросе. Рассмотрим, в чем разница между группировкой и итогами в языке запросов «1С:Предприятие 8».

При построении итогов (как и при группировке) записи, содержащие одинаковые значения полей, по которым нужно рассчитать итоги, собираются вместе, для каждой группы записей вычисляются значения агрегатных функций. Но при этом результат вычислений помещается в итоговые записи и добавляется к детальным записям. То есть детальные записи не сворачиваются, а, наоборот, добавляются новые записи к исходным данным.

Итак, при использовании итогов общее число строк в результате запроса возрастет – к исходным (детальным) записям будут добавлены итоговые (результатирующие) записи. А группировка сворачивает записи, уменьшая тем самым число строк в результате запроса.

Рассмотрим разницу между итогами и группировкой на примере. Будем обращаться запросом к основной таблице движений регистра накопления «ТоварныеЗапасы». Исходный текст запроса без группировок и итогов выглядит следующим образом:

ВЫБРАТЬ

```
ТоварныеЗапасы.Склад КАК Склад,  
ТоварныеЗапасы.Товар КАК Товар,  
ТоварныеЗапасы.ВидДвижения,  
ТоварныеЗапасы.Количество КАК Количество
```

ИЗ

```
РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы
```

Результат запроса выглядит следующим образом:

**Запрос: РегистрНакопления.ТоварныеЗапасы (Записей в результате: 87)**

Склад	Товар	ВидДвижения	Количество
Большой	Sony K3456P	Расход	1
Малый	Ботинки	Расход	3
Малый	Йогурт	Расход	300
Большой	Тапочки	Приход	2
Склад отдела продаж	Veko876N	Приход	100
Средний	Туфли	Приход	2
Средний	Сапоги	Приход	1
Склад отдела продаж	VekoNT02	Приход	200
Малый	Сметана	Приход	2
Средний	Торт	Приход	30
Малый	Veko67NE	Расход	1
Большой	Босоножки	Приход	100
Малый	Сланцы	Приход	2
Малый	Туфли	Приход	2
Большой	Колбаса	Приход	15
Большой	Кроссовки	Приход	20
Большой	Сапоги	Приход	1
Большой	Туфли	Приход	3
Малый	Молоко	Приход	200
Малый	Колбаса	Приход	250
Средний	Хлеб	Приход	300
Большой	Сметана	Приход	11

Перепишем запрос с использованием группировки, просуммировав записи по полю *Количество*:

ВЫБРАТЬ

ТоварныеЗапасы.Склад КАК Склад,  
ТоварныеЗапасы.Товар КАК Товар,  
ТоварныеЗапасы.ВидДвижения,  
СУММА (ТоварныеЗапасы.Количество) КАК Количество

ИЗ

РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы

СТРУППИРОВАТЬ ПО

ТоварныеЗапасы.Склад,  
ТоварныеЗапасы.Товар,  
ТоварныеЗапасы.ВидДвижения

При группировке записей запроса количество строк в результате запроса уменьшится:

**Запрос: РегистрНакопления.ТоварныеЗапасы (Записей в результате: 61)**

Склад	Товар	ВидДвижения	Количество
Средний	Хлеб	Приход	300
Средний	Sony K3456P	Приход	3
Средний	Sony K3456P	Расход	2
Средний	Туфли	Приход	2
Склад отдела продаж	Молоко	Приход	100
Большой	Масло	Приход	1
Малый	Туфли	Приход	2
Малый	Валенки	Приход	40
Малый	Валенки	Расход	3
Большой	Сланцы	Приход	2
Средний	Сапоги	Приход	1
Малый	Ботинки	Приход	50
Малый	Ботинки	Расход	3
Большой	Колбаса	Приход	115
Большой	Колбаса	Расход	60
Большой	Sony K3456P	Приход	7
Большой	Sony K3456P	Расход	1
Малый	Йогурт	Приход	2 000
Малый	Йогурт	Расход	950
Склад отдела продаж	VekoNT02	Приход	200
Большой	Veko67NE	Приход	10
Склад отдела продаж	Veko876N	Приход	100
Малый	Bosch1234	Приход	1

Теперь добавим в запрос расчет итогов:

ВЫБРАТЬ

ТоварныеЗапасы.Склад КАК Склад,  
ТоварныеЗапасы.Товар КАК Товар,  
ТоварныеЗапасы.ВидДвижения КАК ВидДвижения,  
СУММА (ТоварныеЗапасы.Количество) КАК Количество

ИЗ

РегистрНакопления.ТоварныеЗапасы КАК ТоварныеЗапасы

СТРУППИРОВАТЬ ПО

ТоварныеЗапасы.Склад,  
ТоварныеЗапасы.Товар,  
ТоварныеЗапасы.ВидДвижения

ИТОГИ

СУММА (Количество)

ПО

Склад,  
Товар

Результат такого запроса будет следующим:

**Запрос: РегистрНакопления.ТоварныеЗапасы (Записей в результате: 112)**

Склад	Товар	ВидДвижения	Количество
Средний			988
Средний	Хлеб		300
Средний	Хлеб	Приход	300
Средний	Sony K3456P		5
Средний	Sony K3456P	Приход	3
Средний	Sony K3456P	Расход	2
Средний	Туфли		2
Средний	Масло		200
Средний	Сапоги		1
Средний	Торт		230
Средний	Ряженка		200
Средний	Пинетки		50
Склад отдела продаж			600
Склад отдела продаж	Sony K3456P		50
Склад отдела продаж	Sony K3456P	Приход	50
Склад отдела продаж	Молоко		100
Склад отдела продаж	Молоко	Приход	100
Склад отдела продаж	VekoNT02		200

Как видно, при построении итогов по полям *Товар* и *Склад* рассчитывается общее количество движения товаров. Результат вычислений помещается в итоговые строки и добавляется к детальным записям.

Обратите внимание на очень важную вещь. Итоговые записи формируются не в запросе к базе данных, а при помощи обработки полученных данных платформой. В результате запроса, который возвращает сервер СУБД, эти строки физически отсутствуют.



## Видеоурок. Предназначение итогов в запросе. Отличие итогов от группировки

Данный урок записан в видеоформате. Обратитесь к файлу Query-3-Totals-1 на [странице с материалами курса](#).

Изучив этот урок, Вы узнаете:

- Как строятся итоги в запросе на языке «1С:Предприятие 8»
- Когда используются итоги, а когда группировка записей
- Что такое детальные записи, а что такое итоговые записи.

## Предложение ИТОГИ

Предложение ИТОГИ позволяет определить, расчет каких итогов необходим в запросе. При расчете итогов вычисляются значения агрегатных функций по выборкам с одинаковыми значениями полей – контрольных точек. Итоги добавляются в результат запроса как итоговые строки.

После ключевого слова ИТОГИ следует список итоговых полей, перечисленных через запятую. В качестве итоговых полей обычно выступает результат вычисления агрегатных функций, рассчитанный для детальных записей с одинаковыми значениями полей, по которым требуется рассчитать итоги. Затем следует ключевое слово ПО, после которого перечисляется список контрольных точек. Каждая контрольная точка содержит выражение, вычисляемое при выполнении запроса. По каждой комбинации значений этих выражений будут рассчитаны и добавлены в результат запроса итоговые строки.

Например, простой текст запроса для расчета итогов выглядит следующим образом:

ВЫБРАТЬ

```
РасходТовараТовары.Товар КАК Товар,  
РасходТовараТовары.Количество КАК Количество,  
РасходТовараТовары.Сумма КАК Сумма
```

ИЗ

```
Документ.РасходТовара.Товары КАК РасходТовараТовары
```

ИТОГИ

```
СУММА (Количество),  
СУММА (Сумма)
```

ПО

```
Товар
```

Итоговые поля могут рассчитываться необязательно для всех выходных полей запроса. Например, можно рассчитать итоги только по сумме. Это еще одно отличие от группировки, когда все выходные поля необходимо разделить на поля группировки и агрегатные функции.



## Практическое задание №19. Применение итогов в запросах

В ходе выполнения данного задания необходимо получить информацию о средней цене закупа товаров.

Текст задания и его решение скачивайте со [страницы с материалами тренинга](#).

[Перейти к заданию](#)

## Итоги для иерархических справочников

Если контрольная точка в предложении ИТОГИ является ссылкой на справочник, возможен расчет итогов по иерархии справочника. Для этого после такой ссылки нужно указать ключевое слово ИЕРАРХИЯ.

В таком случае текст запроса примет следующий вид:

ВЫБРАТЬ

РасходТовараТовары.Товар КАК Товар,  
РасходТовараТовары.Количество КАК Количество,  
РасходТовараТовары.Сумма КАК Сумма

ИЗ

Документ.РасходТовара.Товары КАК РасходТовараТовары

ИТОГИ

СУММА (Количество) ,  
СУММА (Сумма)

ПО

Товар ИЕРАРХИЯ КАК ИерархияТоваров

Для полей, по которым рассчитываются итоги, можно назначить псевдонимы для последующего обращения к ним из встроеного языка.

**Запрос: Документ.РасходТовара.Товары (Записей в результате: 41)**

Товар	Количество	Сумма
Услуги	1	300
Обувь	15	36 600
Сапоги	4	13 600
Сапоги	2	6 600
Сапоги	2	7 000
Валенки	3	4 500
Валенки	3	4 500
Босоножки	5	12 500
Босоножки	5	12 500
Ботинки	3	6 000
Ботинки	3	6 000
Продукты	1 033	48 910
Йогурт	950	33 000
Колбаса	60	15 000
Сметана	11	550
Молоко	12	360
Электротовары	10	51 600
Телевизоры	3	13 500
Чайники	6	28 100
Пылесосы	1	10 000

На рисунке видно, что в результате запроса рассчитаны итоги по контрольным точкам и итоги по иерархии для контрольных точек.

Можно рассчитать только итоги значений по иерархии, не рассчитывая итоги в контрольных точках. Для этого перед ключевым словом ИЕРАРХИЯ нужно указать ключевое слово ТОЛЬКО:

ВЫБРАТЬ

РасходТовараТовары.Товар КАК Товар,  
РасходТовараТовары.Количество КАК Количество,  
РасходТовараТовары.Сумма КАК Сумма

ИЗ

Документ.РасходТовара.Товары КАК РасходТовараТовары

ИТОГИ

СУММА (Количество),  
СУММА (Сумма)

ПО

Товар ТОЛЬКО ИЕРАРХИЯ КАК ТоварТолькоИерархия

**Запрос: Документ.РасходТовара.Товары (Записей в результате: 27)**

Товар	Количество	Сумма
Услуги	1	300
Доставка	1	300
Обувь	15	36 600
Босоножки	5	12 500
Сапоги	2	6 600
Сапоги	2	7 000
Ботинки	3	6 000
Валенки	3	4 500
Продукты	1 033	48 910
Электротовары	10	51 600
Телевизоры	3	13 500
Sony K3456P	2	9 000
Sony K3456P	1	4 500
Чайники	6	28 100
Пылесосы	1	10 000

На рисунке видно, что итоги были рассчитаны только по иерархии справочника «Товары». Также видно, что итоги рассчитаны по всем уровням иерархии справочника, включая вложенные друг в друга группы. Например, группа *Телевизоры* принадлежит группе *Электротовары*, итоги рассчитаны по каждой из групп справочника. В отличие от предыдущего примера, итогов по отдельным позициям справочника (т.е. по контрольным точкам) в результате запроса нет.

Итоги по иерархии можно построить не только для справочников, но и для планов видов характеристик, планов видов расчетов и т.д.

## Получение итогов по нескольким полям

Если требуется получить итоги по значению нескольких полей, то после ключевого слова ПО нужно перечислить через запятую список контрольных точек. Например, рассчитаем итоги по покупателям и по товарам:

ВЫБРАТЬ

РасходТовараТовары.Ссылка.Покупатель КАК Покупатель,  
РасходТовараТовары.Товар КАК Товар,  
РасходТовараТовары.Количество КАК Количество,  
РасходТовараТовары.Сумма КАК Сумма

ИЗ

Документ.РасходТовара.Товары КАК РасходТовараТовары

ИТОГИ

СУММА (Количество),  
СУММА (Сумма)

ПО

Покупатель,  
Товар

**Запрос: Документ.РасходТовара.Товары (Записей в результате: 41)**

Покупатель	Товар	Количество	Сумма
Магазин "Продукты"		650	21 000
Попов Б.В. ИЧП		314	51 200
Попов Б.В. ИЧП	Йогурт	300	12 000
Попов Б.В. ИЧП	Сапоги	4	13 600
Попов Б.В. ИЧП	Сапоги	2	6 600
Попов Б.В. ИЧП	Сапоги	2	7 000
Попов Б.В. ИЧП	Bosch15	3	8 100
Попов Б.В. ИЧП	Bosch1234	2	5 000
Попов Б.В. ИЧП	Босоножки	5	12 500
Магазин "Мясная лавка"		83	15 910
Шлюзовая ООО		5	28 800
Шлюзовая ООО	Sony K3456P	3	13 500
Шлюзовая ООО	Sony K3456P	2	9 000
Шлюзовая ООО	Sony K3456P	1	4 500
Шлюзовая ООО	Доставка	1	300
Шлюзовая ООО	Доставка	1	300
Шлюзовая ООО	Veko67NE	1	15 000
Шлюзовая ООО	Veko67NE	1	15 000
Магазин "Обувь"		6	10 500
Магазин "Бытовая техника"		1	10 000

Итоги рассчитываются сначала для каждого товара по каждому покупателю, а затем отдельно для каждого покупателя по товарам в целом.