

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 29.07.2022 18:05:30

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d9a44b5c9e1a2bb1c1a

Министерство науки и высшего образования Российской Федерации

Южно-Российский государственный политехнический
университет (НПИ) имени М.И. Платова

С.Н. Широбокова

**БИЗНЕС-АНАЛИТИКА
В ПРИЛОЖЕНИЯХ НА ПЛАТФОРМЕ
«1С:ПРЕДПРИЯТИЕ»:
КОНСПЕКТ ЛЕКЦИЙ**

Учебное пособие

**Новочеркасск
ЮРГПУ(НПИ)
2021**

УДК 004.42 (075.8)
ББК 32.973-018.1я73
Ш64

Широбокова С.Н.

Ш64 Бизнес-аналитика в приложениях на платформе «1С:Предприятие»: конспект лекций/ учебное пособие/ С.Н. Широбокова; Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова. – Новочеркасск: ЮРГПУ(НПИ), 2021.– 116с.

Учебное пособие содержит материалы к лекционным занятиям по дисциплине "Бизнес-аналитика в приложениях на платформе «1С:Предприятие»".

Предназначено для студентов вузов, обучающихся по направлению подготовки «Прикладная информатика» (магистратура). Пособие может быть полезно студентам других компьютерных направлений подготовки, изучающим основы разработки систем бизнес-аналитики на платформе «1С:Предприятие».

УДК 004.42 (075.8)
ББК 32.973-018.1я73

Содержание

Введение	4
Тема 1. Методы и инструментальные средства бизнес-аналитики. Обзор механизма анализа данных и прогнозирования в «1С:Предприятие».	5
Тема 2. Алгоритмы машинного обучения. Типы анализа «Поиск ассоциаций» и "Поиск последовательностей»	15
Тема 3. Алгоритмы машинного обучения. Кластерный анализ в системе «1С:Предприятие»	26
Тема 4. Механизм запросов, табличный способ доступа к данным. Язык запросов в «1С:Предприятии»: роль языка запросов, табличная модель представления данных, структура запроса, выражения в языке запросов	36
Тема 5. Механизм запросов, табличный способ доступа к данным. Язык запросов в «1С:Предприятии»: групповые операции в языке запросов, извлечение данных запросом, вложенные запросы	46
Тема 6. Методы и инструментальные средства бизнес-аналитики. Разработка сложных отчетов с использованием системы компоновки данных	56
Тема 7. Системы на основе аналитики больших данных. Интеграция приложений на платформе «1С:Предприятие» с другими информационными системами: форматы файлов обмена, планы обмена	70
Тема 8. Системы на основе аналитики больших данных. Интеграция приложений на платформе «1С:Предприятие» с другими информационными системами: универсальный механизм обмена данными, обмен данными в распределенных системах	88
Тема 9. Механизм «1С:Аналитика»	101
Заключение	117
Библиографический список	118

ВВЕДЕНИЕ

Отечественные программные продукты, построенные на инновационной технологической платформе «1С:Предприятие 8», ежедневно применяются для повышения эффективности управления и учета более чем на 5 000 000 рабочих мест в организациях различного размера и форм собственности, в России и других странах..

Построение систем бизнес-аналитики на базе технологической платформы «1С:Предприятие» в рамках корпоративных информационных систем весьма актуальная задача. Целью освоения дисциплины «Бизнес-аналитика в приложениях на платформе «1С:Предприятие»» в магистерской подготовке по направлению «Прикладная информатика» является получение обучающимися теоретических представлений о методах, механизмах и технологиях реализации систем бизнес-аналитики и модулей анализа данных в приложениях на платформе «1С:Предприятие», использование которой для разработки российского программного обеспечения соответствует политике импортозамещения, получение практических навыков разработки функционала бизнес-аналитики в корпоративных информационных системах на платформе «1С:Предприятие».

Задачи освоения дисциплины:

- сформировать у обучаемых знания возможностей разработки на базе отечественной технологической платформы «1С:Предприятие» систем бизнес-аналитики и модулей интеллектуального анализа больших данных;
- получить углубленные знания механизма платформы «1С:Предприятие» для анализа данных и прогнозирования для реализации инструментов выявления закономерностей в больших объемах информации корпоративных информационных систем;
- получить знания о механизмах интеграции корпоративных информационных системы на базе платформы «1С:Предприятие» с другими информационными системами, форматы файлов обмена
- развить и закрепить умения применять методы и инструменты анализа данных в решении задач профессиональной деятельности с использованием систем бизнес-аналитики;
- развить и закрепить умения применять язык запросов и систему компоновки данных для разработки сложных аналитических отчетов;
- сформировать профессиональные навыки использования механизма запросов и системы компоновки данных для реализации сложных аналитических отчетов;
- сформировать профессиональные навыки управления разработкой архитектуры комплексных систем бизнес-аналитики со стороны заказчика.

Тема 1. Методы и инструментальные средства бизнес-аналитики. Обзор механизма анализа данных и прогнозирования в «1С:Предприятие»

Общая схема работы механизма анализа и прогнозирования данных. Основные объекты механизма. Типы анализа данных: общая статистика, поиск ассоциаций, поиск последовательностей, кластерный анализ, дерево решений. Модели прогноза.

Механизм анализа данных и прогнозирования (МАДП) платформы «1С:Предприятие» может быть эффективно использован при разработке систем бизнес-аналитики, формировании экономической и аналитической отчетности. Неочевидные закономерности в накопленных в информационной базе данных, которые помогает выявить МАДП, которые могут быть полезны бизнес-аналитикам, экономистам, руководителям для информационной поддержки принятия управленческих решений.

Основные возможности МАДП представлены на рис. 1.1.

Общая схема работы МАДП представлена на рис. 1.2.

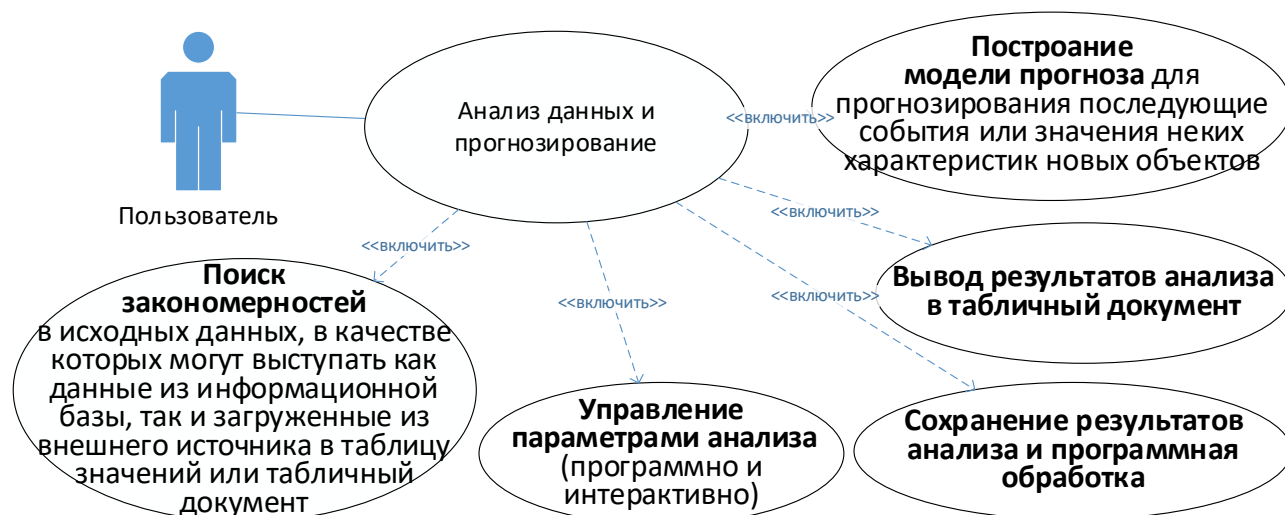


Рис. 1.1. Основные возможности МАДП, предоставляемые пользователю

Механизм МАДП представлен в платформе «1С:Предприятие» набором объектов встроенного языка. Они взаимодействуют друг с другом, имеют ряд параметров для настройки анализа, которые влияют на точность результатов. Разработчик может организовать в приложении интерактивную настройку параметров анализа пользователем или задавать их программно.

В МАДП встроено несколько видов анализа, которые могут быть применены к источнику данных. Результат анализа – это некоторая модель поведения данных. Полученный результат может быть использовать по двум направлениям:

- отображение в итоговом табличном документе;
- сохранение для дальнейшей программной обработки. В частности, на основе результатов анализа может быть создана модель прогноза. Поступающие новые данные подаются на вход такой модели и осуществляется прогноз их поведения в соответствии с имеющейся моделью.

Например, в информационной базе торгового предприятия накапливаются большие данные по продажам товаров клиентам. С помощью вида анализа «Поиск Ассоциаций» МАДП есть возможность проанализировать, какие позиции номенклатуры часто покупаются вместе (в одном чеке). Сохраняем результат анализа в базе данных. Далее разработчиками может быть реализован функционал, позволяющий при создании нового документа продажи строить модель прогноза, на основании сохраненного результата и новых данных – позиций номенклатуры, которые сейчас есть в чеке. На выходе можно получить прогноз в виде списка товаров, которые стоит предложить покупателю, т.к. есть большая вероятность, он их приобретет.



Рис. 1.2. Схема работы МАДП

Типы анализа данных МАДП платформы «1С:Предприятие»:

1) «**Общая статистика**» используется для предварительного исследования анализируемого источника данных, сбора информации о данных, находящихся в исследуемой выборке. Пример результатов представлен на рис. 1.3.

В табл. 1.1. приведены типы данных в статистике.

Таблица 1.1

Типы данных в статистике

Тип переменной	Описание	Пример
1	2	3
Номинальные данные	Данные в виде имени, ярлыка или условного кода	Пол: мужской или женский
Порядковые данные	Данные, у которых интервал между значениями не может быть выражен количественно, но они могут быть ранжированы. Большому значению соответствует большее значение признака.	Степень ожога
Интервальные данные	Данные имеют равный интервал между значениями	Шкала температуры

Продолжение табл. 1.1.

1	2	3
Непрерывные данные	Данные могут принимать любое значение или в определенном диапазоне.	Рост, [150; 190]
Дискретные данные	Данные имеют определенные значения, между которыми четкие границы.	Количество детей, не может быть дробным.

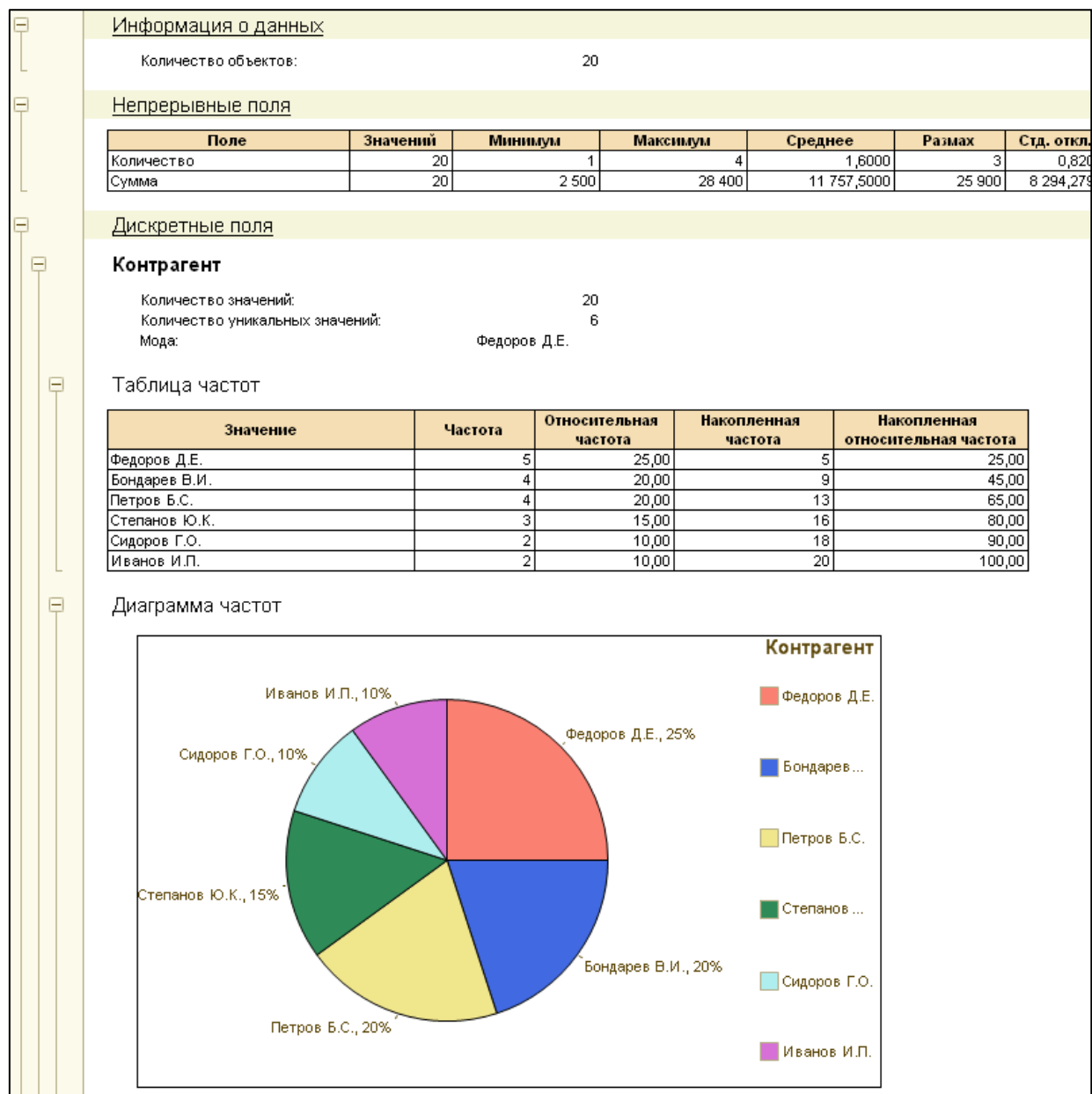


Рис. 1.3. Общая статистика

Расшифруем показатели, которые формирует общая статистика (табл. 1.2).

Таблица 1.2

Показатели анализа «Общая статистика»

Показатель	Описание
<i>Меры центральной тенденции</i>	
Частота	Показывает частоту повторения значения в данных
Среднее	Вычисляется как сумма всех значений деленное на их количество
Мода	чаще всего встречающиеся значения
Медиана	Равна значению, которое делит упорядоченные множество пополам (значение из середины данных). Если значений четное количество, тогда используется среднее от двух ближайших значений
<i>Меры изменчивости данных</i>	
Размах	(максимальное значение – минимальное значение)
Квартили	Три точки (значения) которые делят множество на четыре равные части
Межквартильный размах	отсекаются 25% самых маленьких и больших значений, для остальных считается размах, то есть размах для 50 % данных из середины
Отклонение	Вычисляется как разность конкретного значения от среднего.
Дисперсия	Сумма квадратов отклонений, деленная на их количество.
Стандартное отклонение	Квадратный корень из дисперсии

2) Поиск ассоциаций

Данный тип анализа осуществляет поиск часто встречаемых вместе групп объектов или значений характеристик, а также производит поиск правил ассоциаций. Поиск ассоциаций может использоваться, например, для определения часто приобретаемых вместе товаров, или услуг (комплементарные товары), рис. 1.4.

Этот тип анализа может работать с иерархическими данными, что позволяет, например, находить правила не только для конкретных товаров, но и для их групп. Важной особенностью этого типа анализа является возможность работать как с объектным источником данных, в котором каждая колонка содержит некоторую характеристику объекта, так и с событийным источником, где характеристики объекта располагаются в одной колонке.

Для облегчения восприятия результата предусмотрен механизм отсеечения избыточных правил.

Элемент	Себестоимос...	Выручка	Рентабельность	Доход	Оборачив...	Перио...	Колич...	Средн...
[-] Процент: 1%, Кол-во случаев: 2								
[-] Женская обувь	5 793 382,36	5 826 244,96	33,92	32 862,6	3,61	24 167...	346	2 111,36
[-] Мужская обувь	800	3 250,71	74,46	2 450,71	2,68	217,51	36	90,3
[-] Процент: 1%, Кол-во случаев: 2								
[-] Холодильники однокамерные	1 537,42	2 894,47	36,74	1 357,05	0,04	16 489...	5	512,16
[-] Холодильники двухкамерные	1 552 929,75	1 844 913,63	24,47	291 98...	0,49	40 343,3	124	5 745,26
[-] Процент: 0,5%, Кол-во случаев: 1								
[-] Кондитерские изделия		2 354,1			0,85	51 047...	365	5,02
[-] Бакалея	2 100 000,1	5 527 242,84	59,44	3 424 6...	1,99	567,9	1 320	4 909,69
[-] Процент: 0,5%, Кол-во случаев: 1								
[-] Вентиляторы, пылесосы, кондиционеры	18 551 343,46	34 274 950...	46,78	15 723 ...	4,08	2 349,25	1 047	15 594...
[-] Женская обувь	5 793 382,36	5 826 244,96	33,92	32 862,6	3,61	24 167...	346	2 111,36
[-] Процент: 0,5%, Кол-во случаев: 1								
[-] Вентиляторы, пылесосы, кондиционеры	18 551 343,46	34 274 950...	46,78	15 723 ...	4,08	2 349,25	1 047	15 594...
[-] Телевизоры	220	567,2	61,21	347,2	0,01	86 475...	2	283,6

Рис. 1.4. Представление результатов анализа методом "поиска ассоциаций" в виде групп ассоциированных элементов.

3) Поиск последовательностей

Тип анализа поиск последовательностей позволяет выявлять в источнике данных последовательные цепочки событий. Например, это может быть цепочка товаров или услуг, которые часто последовательно приобретают клиенты.

Этот тип анализа позволяет осуществлять поиск по иерархии, что дает возможность отслеживать не только последовательности конкретных событий, но и последовательности родительских групп.

Набор параметров анализа позволяет специалисту ограничивать временные расстояния между элементами искомым последовательностей, а также регулировать точность получаемых результатов.

4) Кластерный анализ

Кластерный анализ позволяет разделить исходный набор исследуемых объектов на группы объектов, таким образом, чтобы каждый объект был более схож с объектами из своей группы, чем с объектами других групп. Анализируя в дальнейшем полученные группы, называемые кластерами, можно определить, чем характеризуется та или иная группа, принять решение о методах работы с объектами различных групп. Например, при помощи кластерного анализа можно разделить клиентов, с которыми работает компания, на группы, для того, чтобы применять различные стратегии при работе с ними (рис. 1.5).

При помощи параметров кластерного анализа аналитик может настроить алгоритм, по которому будет производиться разбиение, а также может

динамически изменять состав характеристик, учитываемых при анализе, настраивать для них весовые коэффициенты.

Результат кластеризации может быть выведен в дендрограмму — специальный объект, предназначенный для отображения последовательных связей между объектами.

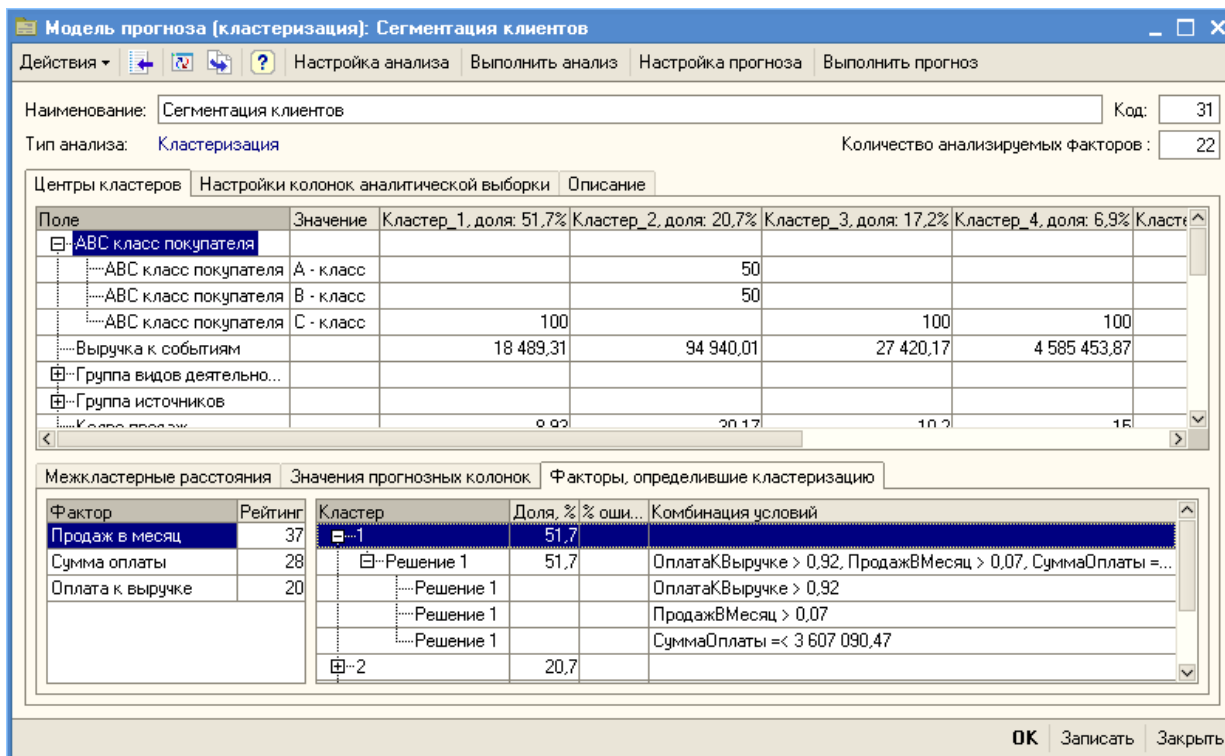


Рис. 1.5. Анализ данных методом кластеризации

5) Дерево решений

Тип анализа дерево решений позволяет построить иерархическую структуру классифицирующих правил, представленную в виде дерева (рис. 1.6).

Для построения дерева решений необходимо выбрать целевой атрибут, по которому будет строиться классификатор и ряд входных атрибутов, которые будут использоваться для создания правил. Целевой атрибут может содержать, например, информацию о том, перешел ли клиент к другому поставщику услуг, удачна ли была сделка, качественно ли была выполнена работа и т. д. Входными атрибутами, для примера, могут выступать возраст сотрудника, стаж его работы, материальное состояние клиента, количество сотрудников в компании и т. п.

Результат работы анализа представляется в виде дерева, каждый узел которого содержит некоторое условие. Для принятия решения, к какому классу следует отнести некий новый объект, необходимо, отвечая на вопросы в узлах, пройти цепочку от корня до листа дерева, переходя к дочерним узлам в случае утвердительного ответа и к соседнему узлу в случае отрицательного.

Набор параметров анализа позволяет регулировать точность полученного дерева.

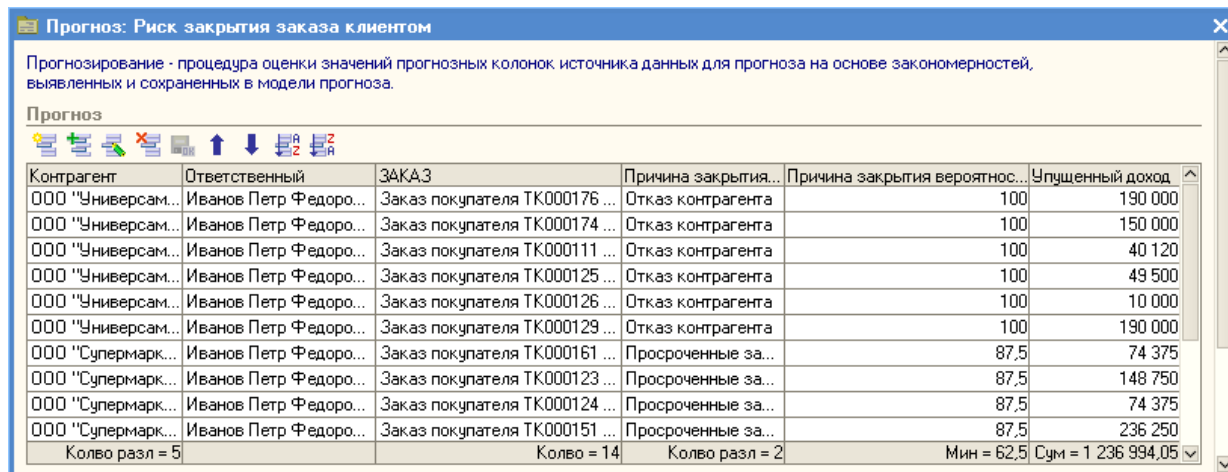
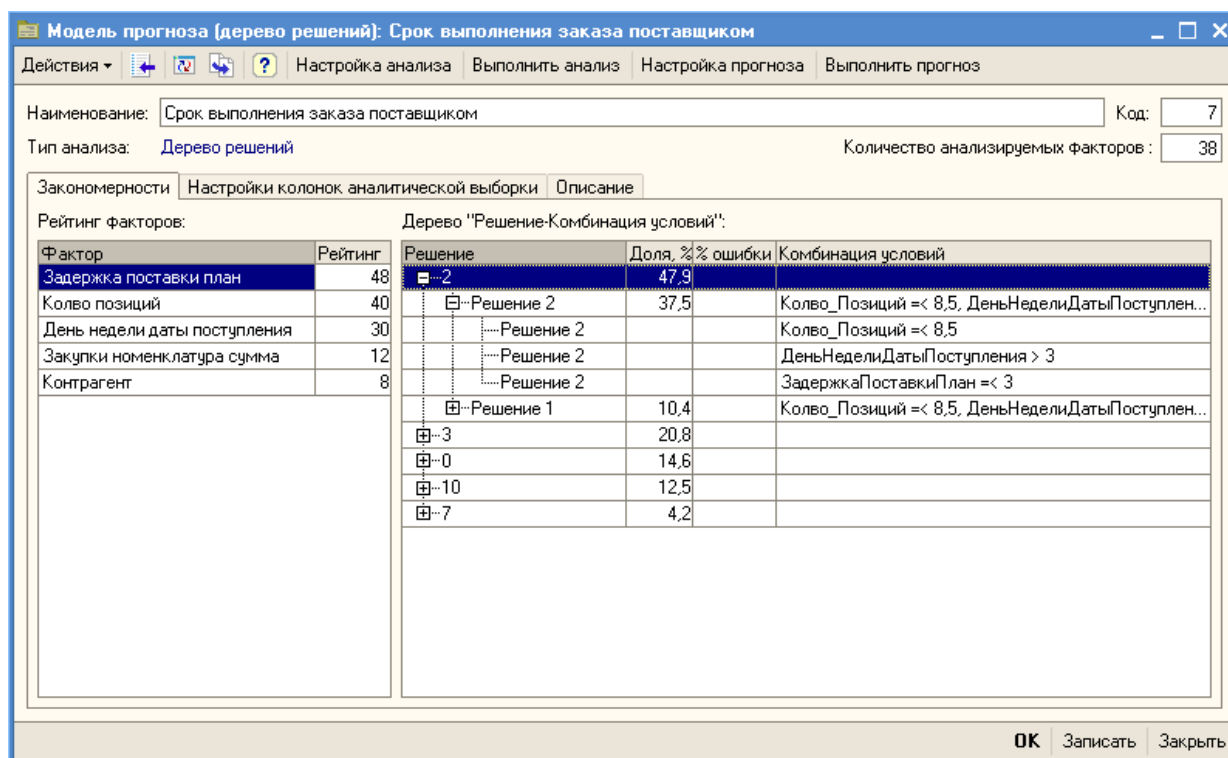


Рис. 1.6. Анализ данных методом «Дерево решений»

Метод «Дерево решений» формирует причинно-следственную иерархию условий, приводящую к определенным решениям. В результате применения этого метода к обучающей выборке создается иерархическая (древовидная) структура правил расщепления вида «Если... то...». Алгоритм анализа (обучения модели) сводится к итеративному процессу вычленения на каждом этапе наиболее значимых условий и переходов между ними. Условия могут иметь как количественный, так и качественный характер и формируют «ветви» этого абстрактного дерева. Его «листву» образуют значения прогнозируемого атрибута (решения), которые, так же как и условия переходов, могут иметь как качественную, так количественную трактовку. Совокупность этих условий, налагаемых на факторы, и структура переходов между ними до конечного решения и образуют модель прогноза.

Данный тип анализа получил наибольшее распространение при оценке исходов различных событийных цепочек и выявлении причинно-следственных связей в выборках. Управление значимостью и достоверностью модели данного алгоритма осуществляется с помощью параметров «Тип упрощения», «Максимальная глубина дерева» и «Минимальное количество элементов в узле».

В качестве результата анализа выборки с помощью алгоритма «Дерево решений» выступают:

- рейтинг факторов, представляющий собой список факторов, оказавших влияние на решение, отсортированный в порядке убывания значимости («цитирования» в узлах дерева);
- сопоставление решений (значений прогнозной колонки) и определивших их условий, иными словами — дерево «Следствие-Причина»;
- дерево «Причина-Следствие», представляющее собой совокупность переходов между условиями, которая определяет то или иное решение (по сути, визуальное представление модели прогноза).

Модели прогноза

Модели прогноза, создаваемые механизмом, представляют собой специальные объекты, которые создаются из результата анализа данных, и позволяют в дальнейшем автоматически выполнять прогноз для новых данных.

Например, модель прогноза поиска ассоциаций, построенная при анализе покупок клиентов, может быть использована при работе с осуществляющим покупку клиентом, для того, чтобы предложить ему товары, которые он с определенной степенью вероятности приобретет вместе с выбранными им товарами.

Типовые бизнес-сценарии использования методов МАДП

Управление взаимоотношениями с клиентами

Сценарий — "Планирование рекламной кампании"

Планирование предстоящей рекламной кампании рассматривается с точки зрения оптимизации распределения выделенного бюджета по рекламным каналам исходя из регионального, продуктового, клиентского и иных показателей целевого сегмента, а также эффективности рекламных каналов в указанных разрезах в некотором, предшествующем планируемому периоде.

- *Алгоритм* — «Кластерный анализ».
- *Прогнозные атрибуты* — доли откликов на рекламный канал условно однородных сегментов, выделенных алгоритмом.
- *Вычисляемые колонки*: доли рекламных каналов в бюджете рекламной кампании с учетом вероятной доли откликов и эффективности (в смысле результирующей выручки) каждого рекламного канала.
- *Пример закономерности*. Клиенты класса А региона П, предпочитающие товарную группу Р, привлечены тем же рекламным каналом, что и клиенты региона Н, предпочитающие товарную группу У.

Управление цепочками поставок

Сценарий — "Оптимизация выбора поставщиков по товарной группе".

Выбор доминирующих поставщиков «первого ряда» для ключевых товарных групп чрезвычайно важен для стабилизации системы логистики в частности и общей системы управления цепочками поставок в целом, уменьшения средней продолжительности цепочек поставок. С другой стороны, более тесная интеграция с основными поставщиками позволяет, как правило, существенно снизить себестоимость товаров. Вместе с тем представляет интерес анализ устойчивых комбинаций поставщиков в различных товарных группах в сравнении с аналитикой по ассоциированным в рамках групп поставщикам. Это дает возможность выявить «пересечения» поставщиков в различных товарных группах и оптимизировать взаимоотношения с ними.

- *Алгоритм* — «Поиск ассоциаций».
- *Прогнозные атрибуты* — устойчивые комбинации поставщиков.
- *Основные факторы* — товарные группы
- *Расшифровка* — аналитика по поставщикам (объем закупок, выручка, условия поставки, оплаты, пессимистичный, оптимистичный, средний сроки выполнения заказа).
- *Пример закономерности.* Устойчивая ассоциация крупного и непредсказуемого поставщика А и предсказуемого среднего поставщика Б в большом количестве товарных групп. Возможно при формировании заказов по конкурентным товарным группам в качестве основного позиционировать среднего поставщика, если объем заказа крупному не превышает некоторого (дающего существенный выигрыш на масштабах) порога.

Управление персоналом

Сценарий — "Профилирование менеджеров отдела продаж по ключевым показателям эффективности".

Определение эффективности менеджеров (удержание, поиск клиентов, эффективность коммуникаций, инкассация условной и безусловной дебиторской задолженности, удельные показатели эффективности на клиента и т. д.) представляет интерес не только с точки зрения формирования системы материального стимулирования менеджеров, но и с точки зрения эффективного нормирования параметров их деятельности.

- *Алгоритм* — «Дерева решений».
- *Прогнозные атрибуты* — ключевые показатели эффективности отдела продаж (количество ключевых клиентов, коэффициенты оттока и привлечения, упущенный доход в месяц, привлеченный доход в месяц, доход в месяц с клиента, суммарные поступления от клиентов и т. д.)
- *Основные факторы* — количество активных клиентов, выручка, доход, удельные показатели на клиента, эффективность коммуникации. В зависимости от прогнозных атрибутов состав факторов может существенно варьироваться.
- *Пример закономерности.* Менеджеры, обеспечивающие лучшие показатели инкассации дебиторской задолженности (отношение поступлений ДС к выручке), имеют коэффициент удержания больше 0,8, коэффициент привлечения больше 0,25, количество одновременно открытых сделок не более

15, но не менее 10, интенсивность событий в день не более 10, но не менее 3, количество активных клиентов в периоде не менее 50, но не более 100.

В качестве примера, иллюстрирующего способность этого типа анализа выявлять причинно-следственные связи, можно привести задачу оптимизации работы отдела продаж. Для ее решения в качестве прогнозируемой величины выберем интересующий показатель эффективности менеджеров по продажам, например удельную доходность на клиента, а в качестве факторов — совокупность данных, потенциально влияющих на результат. Алгоритм определит факторы, оказывающие наибольшее влияние на результат, а также типичные комбинации условий, приводящих к тому или иному результату.

Более того, подсистема "Анализ данных" позволит оценить (спрогнозировать) ожидаемые значения целевого показателя на основании актуальных данных, а также провести прогноз "Что, если?", изменяя подаваемые на вход модели показатели. Результаты анализа и прогноза с помощью деревьев решений дают возможность существенно снизить влияние неопределенности бизнес-окружения на состояние компании, а также решить широкий спектр задач, связанных с выявлением сложных и неочевидных причинно-следственных связей.

Таким образом, механизм анализа данных и прогнозирования – это один из механизмов формирования экономической и аналитической отчетности. Он предоставляет пользователям (экономистам, аналитикам и т.д.) возможность осуществлять поиск неочевидных закономерностей в данных, накопленных в информационной базе.

Механизм анализа данных и прогнозирования позволяет:

- осуществлять поиск закономерностей в исходных данных информационной базы;
- управлять параметрами выполняемого анализа как программно, так и интерактивно;
- осуществлять программный доступ к результату анализа;
- автоматически выводить результат анализа в табличный документ;
- создавать модели прогноза, позволяющие автоматически прогнозировать последующие события или значения неких характеристик новых объектов.

Механизм анализа данных представляет собой набор взаимодействующих друг с другом объектов встроенного языка, что позволяет разработчику использовать его составные части в произвольной комбинации в любом прикладном решении (рис. 1.7).

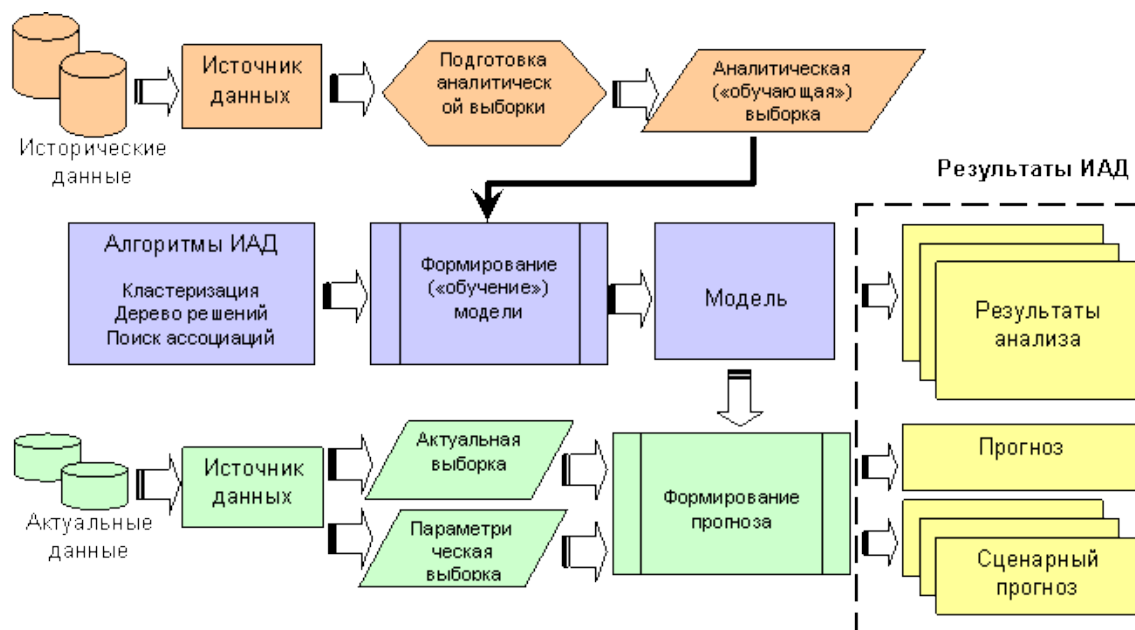


Рис. 1.7. Механизм анализа данных

Встроенные объекты механизма анализа данных позволяют:

- организовать интерактивную настройку параметров анализа пользователем;
- выводить результат анализа в удобной для отображения форме в табличный документ.

Механизм позволяет работать:

- с данными, полученными из информационной базы;
- с данными, полученными из внешнего источника, предварительно загруженными в таблицу значений или табличный документ.

Тема 2. Алгоритмы машинного обучения. Типы анализа «Поиск ассоциаций» и «Поиск последовательностей»

Общая схема выполнения анализа "Поиск ассоциаций". Типы отсеечения правил. Общая схема выполнения анализа "Поиск последовательностей".

Механизм анализа данных и прогнозирования позволяет реализовывать в прикладных решениях различные средства для выявления закономерностей, которые обычно скрываются за большими объемами информации.

Например, проанализировав данные о продажах товаров, можно выявить группы товаров, которые обычно покупаются вместе. В дальнейшем, например, эта информация может использоваться при раскладке товара в розничном магазине. Товары могут располагаться рядом (пришел покупатель за мангалом, увидел рядом жидкость для растопки, угли, мясо, удочки, резиновую лодку... в итоге купил все), могут располагаться в разных углах торгового зала (пришел покупатель за молоком, пока дойдет до хлеба, пройдет через весь магазин и еще чего-нибудь купит).

Другим примером использования механизма анализа данных является прогнозирование поведения контрагента, исходя из имеющихся данных о нем. Проведя такой анализ, можно выяснить, как зависит объем их закупок от территориального расположения, размера компании, времени сотрудничества и прочих показателей. На основании этих зависимостей можно спрогнозировать поведение нового контрагента и выбрать соответствующую стратегию для работы с ним.

Используя возможность построения прогнозов, можно планировать закупочную кампанию. Рассмотрим пример: в прошлом месяце зоомагазин продал 100 морских свинок. Нужно спланировать закупку товара на следующий месяц. Одним из вариантов (очень часто используемых) является ввод поправочного коэффициента на продажи прошлых периодов. Считаем, что поправочный коэффициент (коэффициент повышения спроса) равен 1,5. В итоге при планировании закупок описанным методом будем к следующему месяцу планировать закупку 150 морских свинок. Но если проанализировать, что обычно покупают клиенты после покупки таких питомцев, то можно прийти к совершенно другому выводу.

Воспользовавшись возможностью проведения анализа данных, построения прогноза по этому анализу, можно прийти к выводу, что закупать на следующий месяц нужно корм, различные наполнители, сено и другие «аксессуары».

Общую схему работы механизма анализа и прогнозирования данных можно представить как на рис. 2.1.

Механизм позволяет работать как с данными, полученными из информационной базы, так и с данными, полученными из другого источника, предварительно загруженными в таблицу значений или табличный документ.

Применяя к исходным данным один из видов анализа, можно получить результат анализа. Результат анализа представляет собой некую модель

поведения данных; может быть отображен в итоговом документе или сохранен для дальнейшего использования.

Дальнейшее использование результата анализа заключается в том, что на его основе может быть создана модель прогноза, позволяющая прогнозировать поведение новых данных в соответствии с имеющейся моделью.

Например, можно проанализировать, какие товары приобретаются вместе (в одной накладной), и сохранить созданную на основе данного анализа модель прогноза в базе данных. В дальнейшем при создании очередной накладной ранее сохраненную модель прогноза можно извлечь из информационной базы, подать ей на вход новые данные, содержащиеся в этой накладной, и на выходе получить прогноз – список товаров, которые очередной клиент тоже приобретет (с определенной долей вероятности), если их ему предложить.

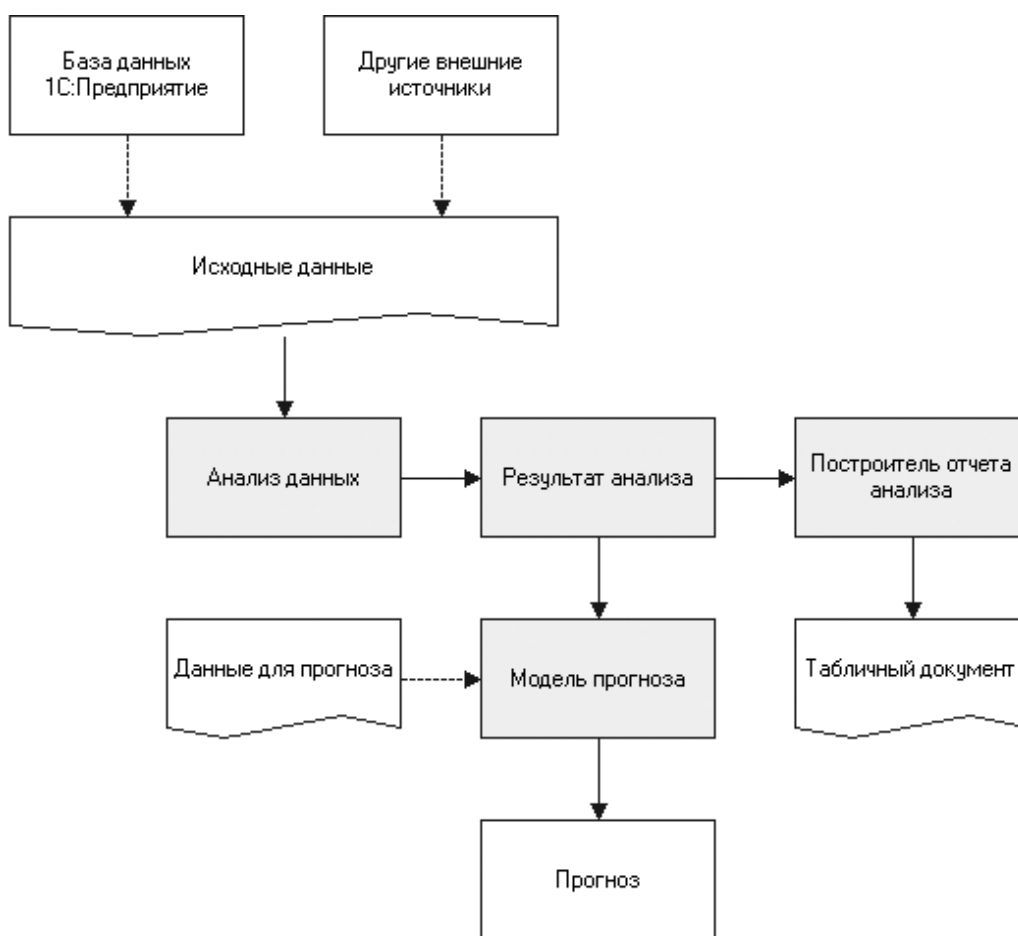


Рис. 2.1. Общая схема работы механизма анализа и прогнозирования данных

Основные объекты механизма

Схематично взаимосвязь основных объектов механизма анализа данных и прогнозирования можно показать следующим образом (см. рис. 2.2).

АнализДанных – объект, непосредственно выполняющий анализ данных. Ему устанавливается источник данных, задаются параметры и исходные данные. Результатом работы данного объекта является результат анализа данных, причем для каждого типа анализа существует свой объект для работы с результатом анализа:

- РезультатАнализаДанныхОбщаяСтатистика,
- РезультатАнализаДанныхПоискАссоциаций,
- РезультатАнализаДанныхПоискПоследовательностей,
- РезультатАнализаДанныхДеревоРешений,
- РезультатАнализаДанныхКластеризация.



Рис. 2.2. Взаимосвязь основных объектов механизма анализа данных и прогнозирования

Настройка колонок анализа данных – коллекция входных колонок анализа данных. Для каждой колонки указывается тип данных, содержащихся в ней, роль, выполняемая колонкой, дополнительные настройки, зависящие от типа производимого анализа.

Параметры анализа данных – набор параметров производимого анализа данных. Состав параметров зависит от типа анализа. Например, для кластерного анализа указывается количество кластеров, на которые необходимо разбить исходные объекты, тип измерения расстояния между объектами и т. п.

Источник данных – исходные данные для анализа. В качестве источника данных может выступать результат запроса, область ячеек табличного документа, таблица значений.

Результат анализа данных – специальный объект, содержащий информацию о результате анализа. Для каждого вида анализа предусмотрен свой результат. Например, результатом анализа данных Дерево решений будет объект типа РезультатАнализаДанныхДеревоРешений. В дальнейшем результат может быть выведен в табличный документ при помощи построителя отчета анализа данных, может быть выведен посредством программного доступа к его содержимому, может быть использован для создания модели прогноза. Любой результат анализа данных может быть сохранен для последующего использования.

Модель прогноза – специальный объект, позволяющий выполнять прогноз на основании входных данных (выборка для прогноза, настройки колонок выборки и результата, результат анализа). Тип модели прогноза зависит от типа результата анализа данных. Например, модель, созданная для типа анализа Поиск ассоциаций, будет иметь тип МодельПрогнозаПоискАссоциаций. Такая модель сможет выдавать прогнозы типа: т. к. данный покупатель купил заданный набор товаров, то с определенной вероятностью он должен купить и другой набор товаров. На вход модели прогноза передается источник данных для прогноза. Результатом является таблица значений, содержащая прогнозируемые значения.

Настройка входных колонок – набор специальных объектов, показывающих соответствие между колонками модели прогноза и колонками выборки прогноза. Например, колонке модели прогноза с именем Товар может соответствовать колонка выборки Номенклатура.

Настройка колонок результата – позволяет управлять тем, какие колонки будут помещены в результирующую таблицу модели прогноза. Например, для поиска ассоциаций мы можем вывести в результат номенклатуру, которую, скорее всего, приобретет клиент, и вероятность подобной покупки.

Колонки результата – таблица значений, состоящая из колонок, указанных в настройках результирующих колонок, и содержащая прогнозируемые данные. Конкретное содержимое определяется типом анализа.

Типы анализа данных.

В механизме анализа данных и прогнозирования реализовано несколько типов анализа данных:

Рассмотрим более подробно два из них.

Тип анализа «Поиск ассоциаций» (рис. 2.3).

Данный тип анализа осуществляет поиск часто встречаемых вместе комбинаций объектов или значений характеристик. С его помощью можно определять группы одновременно закупаемых товаров, выявлять наиболее привлекательные источники информации (в процессе «оптимизации» затрат на них) и т. п. Схематично процесс проведения анализа Поиск ассоциаций можно представить следующим образом:

В качестве источника данных может использоваться результат запроса, таблица значений или область ячеек табличного документа. С точки зрения данного типа анализа колонки источника можно разделить на следующие.

- *НеИспользуемая* – игнорируются анализом;
- *Объект* – данные из этой колонки используются как объекты (или события) проводимого анализа. Исходя из значения данной колонки, значения другой колонки (Элемент) относятся к одной ассоциируемой группе;
- *Элемент* – данные из этой колонки используются для получения устойчивых групп значений, построения ассоциативных правил.

Кроме настройки типов колонок, на результат проводимого анализа влияют следующие параметры анализа:

- **МинимальныйПроцентСлучаев** – определяется минимальный процент случаев, в которых должна встречаться комбинация элементов. Группы, у которых данное значение меньше указанного, не попадают в результат анализа;
- **МинимальнаяДостоверность** – показывает минимальное значение процента случаев, когда правило соблюдается;
- **МинимальнаяЗначимость** – группы, у которых данное значение меньше указанного, не попадают в результат анализа;
- **ТипОтсеченияПравил** – один из вариантов системного перечисления ТипОтсеченияПравилАссоциации: Избыточные (отсекать избыточные правила), Покрытые (отсекать правила, покрытые другими правилами).

В результате выполнения анализа получаем:

- информацию о данных (количество объектов, количество элементов, среднее количество элементов в объекте, количество найденных групп, количество найденных правил ассоциаций);

- найденные группы элементов – указывается состав группы, количество случаев, процент случаев, в которых эта группа встречается;

- найденные ассоциативные правила – указывается исходный состав элементов, следствие (состав элементов), процент случаев, достоверность, значимость правила.



Рис. 2.3. Схема процесса проведения анализа «Поиск ассоциаций»

Пример. Задача: **определить состав одновременно закупаемых товаров**

Признаком, по которому данные относятся к одной группе, будем считать значение регистратора (номенклатура, указанная в одном документе, считается закупленной одновременно). То есть Регистратор будет объектом анализа, Номенклатура – элементом.

Для проведения анализа будем использовать следующий фрагмент кода (рис. 2.4):

```

&НаКлиенте
Процедура ПоискАссоциаций(Команда)
    Результат = АнализПоискАссоциаций();
КонецПроцедуры
&НаСервереБезКонтекста
Функция АнализПоискАссоциаций()
    Анализ = Новый АнализДанных;
    Анализ.ТипАнализа = Тип("АнализДанныхПоискАссоциаций");
    Запрос = Новый Запрос;
    Запрос.Текст = "
        |ВЫБРАТЬ
        |Продажи.Регистратор,
        |Продажи.Номенклатура
        |ИЗ
        |РегистрНакопления.Продажи КАК Продажи";
    Анализ.ИсточникДанных = Запрос.Выполнить();
    // Строка приводится в качестве примера,
    // такое значение типа колонки по умолчанию.
    Анализ.НастройкаКолонок.Номенклатура.ТипКолонки =
ТипКолонкиАнализаДанныхПоискАссоциаций.Элемент;
    // Строка приводится в качестве примера,
    // такое значение типа отсечения по умолчанию.
    Анализ.Параметры.ТипОтсеченияПравил.Значение = ТипОтсеченияПравилАссоциации.Избыточные;
    РезультатАнализа = Анализ.Выполнить();
    Построитель = Новый ПостроительОтчетаАнализаДанных();
    Построитель.Макет = Неопределено;
    Построитель.ТипАнализа = Тип("АнализДанныхПоискАссоциаций");
    ТабДок = Новый ТабличныйДокумент;
    Построитель.Вывести(РезультатАнализа, ТабДок);
    Возврат ТабДок;
КонецФункции

```

Рис. 2.4. Код проведения анализа Поиск ассоциаций

Результат анализа будет выглядеть следующим образом (рис. 2.5):

Информация о данных

Количество элементов:	12
Количество объектов:	11
Среднее количество элементов в объекте:	1,82

Результат анализа

Найдено часто встречаемых групп:	1
Найдено ассоциативных правил:	2

Рис. 2.5. Результат анализа Поиск ассоциаций

В выборе используются данные из одиннадцати документов (ссылка содержится в поле Регистратор), количество различных номенклатурных позиций равно двенадцати (рис. 1.4).

Найдена следующая группа товаров (рис. 2.6). Вся группа встречается в документе только в двух случаях из одиннадцати (это и показано в колонках Количество случаев и Процент случаев). Получены следующие ассоциативные правила (рис. 2.7).

Разберем второе из них. В двух случаях из одиннадцати в документе вместе с позицией Стол кухонный раскладной встречалась позиция Табурет прямоугольный. Исходя из этого, был рассчитан процент случаев: $(2 / 11 * 100 = 18,18 \%)$.

Различные номенклатурные позиции

Номенклатура
Стол кухонный раскладной
Табурет круглый
Диван «УЮТ»
Диван «Джинс»
Кресло «Джинс»
Стол «Kitchen» 0.9x1.7
Диван «Комфорт»
Стул «Summer»
Табурет прямоугольный
Кресло «УЮТ»
Шкаф «Wardrobe»
Стол обеденный

Рис. 2.6.

Достоверность была рассчитана следующим образом: обе номенклатурные позиции закупались в двух случаях, товарная позиция Стол кухонный раскладной встречалась в покупках 3 раза. Исходя из этого, достоверность равна: $2 / 3 * 100 = 66,67 \%$.

Значимость определяется как отношение достоверности правила к проценту нахождения Табурет прямоугольный в закупаемых товарах. Эта позиция встречается в двух документах из одиннадцати (18,18 %). Значимость равна: $66,67 \% / 18,18 \% = 3,67$.

Типы отсечения правил

Рассмотрим такой важный параметр данного типа анализа, как ТипОтсеченияПравил. Возможные значения отсечения содержатся в системном перечислении ТипОтсеченияПравилАссоциации, их состав:

Покрытые,
Избыточные.

Перед тем как перейти к рассмотрению особенностей вариантов отсечения, рассмотрим несколько общих моментов, применимых к правилам ассоциации.

Ассоциативные правила

Исходный набор	Следствие	Процент случаев	Достоверность	Значимость
1		18,18	100,00	3,67
Номенклатура = Табурет прямоугольный	Номенклатура = Стол кухонный раскладной			
2		18,18	66,67	3,67
Номенклатура = Стол кухонный раскладной	Номенклатура = Табурет прямоугольный			

Рис. 2.7. Ассоциативные правила

Любое правило состоит из предпосылки и следствия. Например:

- Предпосылка: Если купили Товар № 1.
- Следствие: То купят Товар № 2.

При этом не нужно забывать, что следствие наступает с определенной достоверностью. При отсечении правил вероятностные характеристики могут рассматриваться, а могут и игнорироваться (важно только содержание правила).

Отсечение покрытых правил

Рассмотрим вариант отсечения **Покрытые**.

Правило может быть покрыто как по предпосылке, так и по следствию.

Например:

- Правило № 1: Если купили товар № 1 и № 3, То купят товар № 2.
- Правило № 2: Если купили товар № 1, То купят товар № 2.

В этом случае правило № 1 будет считаться покрытым, т. к. предпосылка первого правила получается «избыточной» по отношению к предпосылке второго правила.

Пример покрытия по следствию:

- Правило № 1: Если купили товар № 1, То купят товар № 2, № 3.
- Правило № 2: Если купили товар № 1, То купят товар № 3.

Правило № 2 будет покрыто по следствию, так как следствие правила № 1 более полное.

Отсечение избыточных правил

Покрытие не учитывает вероятностных характеристик правил, они учитываются в случае, если используется вариант отсечения избыточные.

Правило будет считаться избыточным по предпосылке, если оно покрыто по предпосылке и его достоверность равна достоверности покрывающего правила. Например:

- Правило № 1: Если купили товар № 1 и № 3, То купят товар № 2 с достоверностью 75 %.
- Правило № 2 Если купили товар № 1, То купят товар № 2 с достоверностью 75 %.

Правило № 1 избыточно по отношению к правилу № 2 (оно содержит дополнительное условие, не вносящее «возмущения» в вероятностные характеристики правила).

Правило № 1 будет считаться избыточным по следствию, если количество случаев данного правила равно количеству случаев покрывающего правила.

- Правило № 1: Если купили товар № 1, То купят товар № 2, № 3 в трех случаях.
- Правило № 2: Если купили товар № 1, То купят товар № 3 в трех случаях.

Правило № 2 будет считаться избыточным по отношению к правилу № 1, так как оно содержит более простое следствие с теми же вероятностными характеристиками.

Тип анализа «Поиск последовательностей». Данный тип анализа позволяет выявить цепочки возникающих событий (шаблоны последовательностей). Он может использоваться тогда, когда одним из важных анализируемых показателей является последовательность наступления событий во времени. Например, можно выявить последовательность товаров, которые покупаются друг за другом в течение какого-либо определенного промежутка времени и т. п. Схематично процесс проведения анализа Поиск последовательностей можно представить следующим образом (рис. 2.8).

В качестве источника данных может использоваться результат запроса, таблица значений, область ячеек табличного документа. В табл. 2.2 приведен пример выборки данных для анализа.

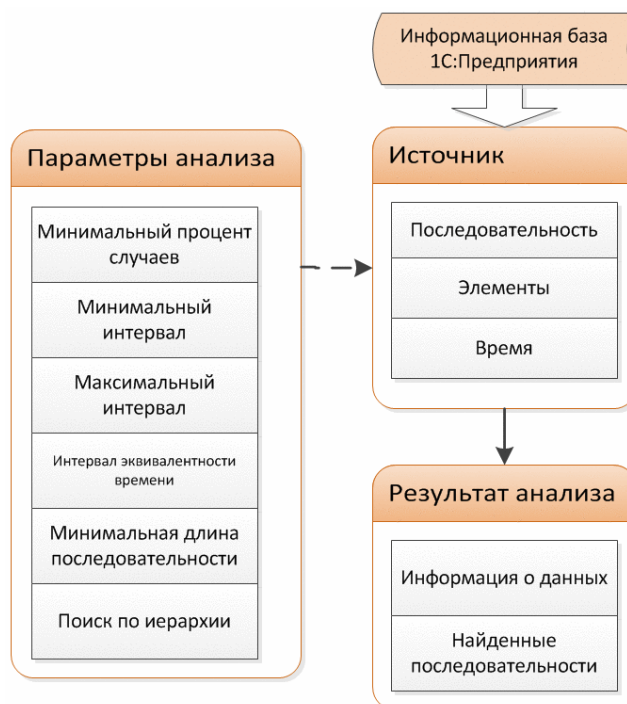


Рис. 2.8. Схема анализа «Поиск последовательностей»

Таблица 2.2

Выборка данных

Контрагент	Первая покупка	Вторая покупка	Третья покупка	Интервал
Бондарев В.И.	Стол кухонный раскладной	Диван «УЮТ»	Кресло «УЮТ»	25 дней, 31 день
	Табурет круглый			
Иванов И.П.	Диван «Джинс»			
	Кресло «Джинс»			
Петров Б.С.	Стол «Kitchen» 0.9x1.7	Кресло «УЮТ»		43 дня
	Стул «Summer»			
	Диван «УЮТ»			
Сидоров Г.О.	Стол кухонный раскладной			
	Табурет прямоугольный			
Степанов В.К.	Стол кухонный раскладной			
	Табурет прямоугольный			
	Стол обеденный			
Федоров Д.Е.	Стол «Kitchen» 0.9x1.7	Шкаф «Wardrobe»	Стул «Summer»	58 дней, 29 дней
	Диван «Комфорт»		Табурет круглый	

С точки зрения данного типа анализа колонки источника можно разделить на следующие:

- НеИспользуемая – игнорируются анализом;
- Последовательность – данные из этой колонки используются для анализа как объект события последовательности. По значению данной колонки анализ и ассоциирует данные с одной цепочкой событий;
- Элемент – данные из этой колонки используются как элементы последовательности;
- Время – именно по данной колонке определяется время наступления события. Наличие данной колонки обязательно при проведении данного типа анализа. Кроме настройки типов колонок, на результат проводимого анализа влияют следующие параметры анализа:
- МинимальныйПроцентСлучаев – минимальный процент последовательностей, в которых наблюдается найденный шаблон последовательности;
- МинимальныйИнтервал – признак установки минимального интервала последовательности (должна быть установлена единица измерения интервала, кратность);
- МаксимальныйИнтервал – признак установки максимального интервала последовательности (должна быть установлена единица измерения интервала, кратность);
- ИнтервалЭквивалентностиВремени – признак установки интервала эквивалентности времени (должна быть установлена единица интервала эквивалентности времени, ее кратность);
- МинимальнаяДлина – минимальная длина искомым последовательностей;
- ПоискПоИерархии – признак осуществления поиска по иерархии (распространяется на колонки с типом Элемент).

Ряд свойств оперирует таким понятием, как ТипЕдиницыИнтервалаВремениАнализаДанных. Данное системное перечисление содержит следующие значения (рис. 2.9).

Значения системного перечисления «ТипЕдиницыИнтервалаВремениАнализаДанных»

Секунда	
Минута	ТекущаяМинута
Час	ТекущийЧас
День	ТекущийДень
Неделя	ТекущаяНеделя
Декада	ТекущаяДекада
Месяц	ТекущийМесяц
Квартал	ТекущийКвартал
Полугодие	ТекущееПолугодие
Год	ТекущийГод

Рис. 2.9. Значения системного перечисления ТипЕдиницыИнтервалаВремениАнализаДанных

Основным результатом анализа являются найденные шаблоны последовательностей. Эти шаблоны содержат следующую информацию:

- состав шаблона последовательности;
- количество случаев, в которых наблюдалась данная последовательность;
- максимальные интервалы между событиями (в случае, если событий 2, то интервал один);
- минимальные интервалы между событиями (в случае, если событий 2, то интервал один);
- процент случаев, когда данная последовательность выполнялась;
- средние интервалы между событиями (в случае, если событий 2, то интервал один).

```

&НаКлиенте
Процедура ПоискПоследовательностей(Команда)
    Результат = АнализПоискПоследовательностей();
КонецПроцедуры
&НаСервереБезКонтекста
Функция АнализПоискПоследовательностей()
    Анализ = Новый АнализДанных;
    Анализ.ТипАнализа = Тип("АнализДанныхПоискПоследовательностей");
    Запрос = Новый Запрос;
    Запрос.Текст = "
        |ВЫБРАТЬ
        |Продажи.Контрагент,
        |Продажи.Номенклатура,
        |Продажи.Период
        |ИЗ
        |РегистрНакопления.Продажи КАК Продажи";
    Анализ.ИсточникДанных = Запрос.Выполнить();
    Анализ.НастройкаКолонок.Период.ТипКолонки =
ТипКолонкиАнализаДанныхПоискПоследовательностей.Время;
    РезультатАнализа = Анализ.Выполнить();
    Построитель = Новый ПостроительОтчетаАнализаДанных();
    Построитель.Макет = Неопределено;
    Построитель.ТипАнализа = Тип("АнализДанныхПоискПоследовательностей");
    ТабДок = Новый ТабличныйДокумент;
    Построитель.Вывести(РезультатАнализа, ТабДок);
    Возврат ТабДок;
КонецФункции

```

Рис. 2.10. Пример кода анализа «Поиск последовательностей»

Пример кода анализа «Поиск последовательностей» приведены на рис. 2.10.

Данные из колонки Контрагент будут определять принадлежность к конкретной цепочке событий, т. е. они определяют последовательность анализа. Номенклатура будет являться элементом получаемой последовательности. Для проведения анализа может использоваться фрагмент кода, аналогичный приведенному ниже (рис. 2.11).

Параметры анализа, установленные по умолчанию (рис. 2.12).

Параметры анализа

Минимальный процент случаев:	10
Минимальный интервал:	---
Максимальный интервал:	---
Интервал эквивалентности времени:	---
Минимальная длина последовательности:	2

Рис. 2.11. Параметры анализа

Информация о данных

Количество элементов:	12
Количество последовательностей:	6

Результат анализа

Найдено последовательностей:	2
------------------------------	---

Рис. 2.12. Результат анализа

После проведения анализа получены следующие общие данные (рис. 2.12):

Количество элементов равно двенадцати. Ровно столько номенклатурных позиций встречается в приведенной выборке данных. Найдены две последовательности (рис. 2.13).

Первая последовательность встречается в двух случаях из пяти. Исходя из этого, процент случаев равен 40 %. Так как глубина последовательности равна 2, существует по одному значению каждого из приводимых интервалов.

Последовательности

№	Количество случаев	Процент случаев	Средний интервал	Минимальный интервал	Максимальный интервал
Состав					
1	2	33,33			
Номенклатура = Диван "УЮТ"					
Номенклатура = Кресло "УЮТ"			1 мес. 7 д.	1 мес. 1 д.	1 мес. 14 д.
2	1	16,67			
Номенклатура = Стол "Kitchen" 0.9x1.7					
Номенклатура = Стул "Summer"			2 мес. 28 д.	2 мес. 28 д.	2 мес. 28 д.

Рис. 2.13. Найденные последовательности

Тема 3. Алгоритмы машинного обучения. Кластерный анализ в системе «1С:Предприятие»

Кластерный анализ в системе «1С:Предприятие». Методы кластеризации: Близняя связь, Дальняя связь, Центр тяжести, «k средних». Вывод данных в дендрограмму.

Тип анализа «Кластеризация»

Кластерный анализ позволяет разделить исходный набор исследуемых объектов на группы объектов таким образом, чтобы каждый объект был более схож с объектами из своей группы, чем с объектами других групп. Анализируя в дальнейшем полученные группы, называемые кластерами, можно определить, чем характеризуется та или иная группа, принять решение о методах работы с объектами различных групп. Например, при помощи кластерного анализа можно разделить клиентов, с которыми работает компания, на группы, для того, чтобы применять различные стратегии при работе с ними.

При помощи параметров кластерного анализа пользователь может настроить алгоритм, по которому будет производиться разбиение, а также может динамически изменять состав характеристик, учитываемых при анализе, настраивать для них весовые коэффициенты.

Результат кластеризации может быть выведен в дендрограмму - специальный вид диаграммы, предназначенный для графического отображения результатов кластерного анализа.

Кластерный анализ – математическая процедура многомерного анализа, позволяющая на основе множества показателей, характеризующих ряд объектов, сгруппировать их в кластеры таким образом, чтобы объекты, входящие в один кластер, были более однородными, сходными, по сравнению с объектами, входящими в другие кластеры.



Рис. 3.1. Схема механизма проведения кластерного анализа

В основе данного анализа лежит вычисление расстояния между объектами. Именно исходя из расстояний между объектами и производится их группировка по кластерам. Определение расстояния может проводиться разными способами (по разным метрикам). Поддерживаются следующие метрики:

- Евклидова метрика,
- Евклидова метрика в квадрате,
- Метрика города,
- Метрика доминирования.

После определения расстояний между объектами может использоваться один из нескольких алгоритмов распределения объектов по кластерам. Поддерживаются следующие методы кластеризации:

- Ближняя связь,
- Дальняя связь,
- k-средних,
- Центр тяжести.

Схематично механизм проведения кластерного анализа можно представить следующим образом (рис. 3.1).

На вход объекту **АнализДанных** подается источник данных. В качестве источника может выступать результат запроса, таблица значений, область ячеек табличного документа. Колонки источника определяются как входные либо неиспользуемые. Следует отметить, что все значения колонок содержатся в системном перечислении **ТипКолонкиАнализаДанныхКластеризация**. В этом

```

&НаКлиенте
Процедура КластерныйАнализ(Команда)
    Результат = АнализКластеризация();
КонецПроцедуры
&НаСервереБезКонтекста
Функция АнализКластеризация()
    Анализ = Новый АнализДанных;
    Анализ.ТипАнализа = Тип("АнализДанныхКластеризация");
    Группа = Справочники.Контрагенты.НайтиПоНаименованию("Юридические лица");
    Запрос = Новый Запрос;
    Запрос.Текст = "
        |ВЫБРАТЬ
        |Контрагенты.Ссылка,
        |Контрагенты.КоличествоРозничныхТочек,
        |Контрагенты.КоличествоАвтомобилей,
        |Контрагенты.ВремяРаботыОрганизации,
        |Контрагенты.ВремяЗаключенияДоговора,
        |Контрагенты.ВидДоговора,
        |Контрагенты.ПрекращениеОтношений
        |ИЗ
        |Справочник.Контрагенты КАК Контрагенты
        |ГДЕ
        |(НЕ Контрагенты.ЭтоГруппа И Контрагенты.Родитель = &Родитель)";
    Запрос.УстановитьПараметр("Родитель", Группа);
    Анализ.ИсточникДанных = Запрос.Выполнить();
    // Выбор метрики.
    Анализ.Параметры.МераРасстояния.Значение =
ТипМерыРасстоянияАнализаДанных.ЕвклидоваМетрикаВКвадрате;
    // Выбор метода кластеризации.
    Анализ.Параметры.МетодКластеризации.Значение = МетодКластеризации.КСредних;
    РезультатАнализа = Анализ.Выполнить();
    Построитель = Новый ПостроительОтчетаАнализаДанных();
    Построитель.Макет = Неопределено;
    Построитель.ТипАнализа = Тип("АнализДанныхКластеризация");
    ТабДок = Новый ТабличныйДокумент;
    Построитель.Вывести(РезультатАнализа, ТабДок);
    Возврат ТабДок;
КонецФункции

```

Рис. 3.2. Код проведения кластерного анализа

перечислении значений больше (не только неиспользуемые и входные), но другие значения используются при построении прогнозов.

Анализ производится в соответствии с установленными параметрами анализа. В качестве примера, иллюстрирующего возможность проведения кластерного анализа, будем использовать следующий фрагмент кода (рис. 3.2).

Запрос выполняется по справочнику **Контрагенты**. По условию запроса выбираются только детальные записи справочника из группы **Юридические лица**. Выполнение указанного кода приведет к тому, что в качестве начальных установок анализа данных будут определены следующие значения (часть установлена явно, часть – по умолчанию), рис. 3.3.

Состав колонок определен, исходя из состава полей выборки запроса. По умолчанию они определены с равным весом. Для типов **Число** и **Дата** определен вид данных **Непрерывные**, для остальных типов – **Дискретные**. При необходимости изменить параметры колонок это можно сделать по аналогии с приведенным фрагментом:

Анализ.НастройкаКолонок.КоличествоАвтомобилей.ДополнительныеПараметры.Вес= 2;

В данной строке для колонки **КоличествоАвтомобилей** увеличен вес.

Выборка данных, для которых будет произведен анализ, имеет следующее наполнение (табл. 3.1).

Таблица 3.1.

Исходная выборка данных

Контрагент	Количество розничных точек	Количество автомобилей	Время работы организации	Время заключения договора	Вид договора	Состояние взаимоотношений
1	2	3	4	5	6	7
ЗАО МДК	1	0	МеньшеГода	МеньшеГода	Дилер	Несоблюдение-Договора
ООО «Лидер»	15	4	ОтТрех-ДоДесяти-Лет	МеньшеГода	Дистрибьютер	Прекращение-Контрагентом

Кластерный анализ

Параметры анализа

Количество искомых кластеров: 3
 Стандартизация: Стандартизировать
 Мера расстояния: Евклидова метрика в квадрате
 Метод кластеризации: Метод k-средних

Колонки источника данных

Входные колонки

Имя колонки	Тип данных	Вес
Ссылка	Дискретный	1
КоличествоРозничныхТочек	Непрерывный	1
КоличествоАвтомобилей	Непрерывный	1
ВремяРаботыОрганизации	Дискретный	1
ВремяЗаключенияДоговора	Дискретный	1
ВидДоговора	Дискретный	1
ПрекращениеОтношений	Дискретный	1

Рис. 3.3. Начальные установки анализа данных

Продолжение табл. 3.1

1	2	3	4	5	6	7
ООО «Дуб-Мебель»	1	10	ОтТрех-ДоДесяти-Лет	ОтГода-ДоТрех	Дистри-бьютер	Прекра-щениеКонтр-агентом
ИП Огарьков С.К.	1	1	ОтГода-ДоТрех	Мень-шеГода	Дилер	Прекра-щениеКонтр-агентом
Магазин №23	1	1	Свыше-Десяти-Лет	ОтТрех-ДоДесяти-Лет	Постоян-ный-Партнер	НеПрекра-щены
ООО «Быстрая-Пицца»	3	2	Мень-шеГода	Мень-шеГода	Постоян-ный-Партнер	НеПрекра-щены
ООО «Василиса»	7	3	ОтТрех-ДоДесяти-Лет	ОтГода-ДоТрех	Постоян-ный-Партнер	Прекра-щениеКонтр-агентом
ООО «Силай»	2	2	Свыше-Десяти-Лет	ОтТрех-ДоДесяти-Лет	Постоян-ный-Партнер	НеПрекра-щены
ИП Добров К.Л.	0	1	Мень-шеГода	Мень-шеГода	Дилер	НеПрекра-щены

Результат кластерного анализа данных из в табл. 3.1, приведен на рис. 3.4. Поскольку мы не настраивали программно явно параметр **ТипЗаполненияТаблицы**, полученные результаты отражают только информацию о найденных кластерах (количество, центра, расстояния между ними).

Для того чтобы в результате анализа увидеть распределение объектов по кластерам, необходимо перед выполнением анализа (но после определения его типа) определить следующую строку кода:

```
Анализ.Параметры.ТипЗаполненияТаблицы.Значение =
ТипЗаполненияТаблицыРезультатаАнализаДанных.ИспользуемыеПоля;
```

При анализе могут использоваться входные колонки не только непрерывного типа (для этого типа очевидно понятие «расстояния»), и колонки дискретных типов (ссылки на справочники, значения перечислений и т. п.).

Используемые метрики.

Разберемся с метриками, которые могут использоваться при проведении кластерного анализа.

<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<u>Информация о данных</u>	Количество объектов:	9
	<u>Результат анализа</u>	Найдено кластеров:	3
	<u>Кластеры</u>		
	№ кластера	Количество объектов	Процент
	1	6	66,67
	2	2	22,22
	3	1	11,11

Центры кластеров

Поле \ Кластер	1	2	3
Ссылка = ЗАО МДК	0,1667		
Ссылка = ИП Огарьков С.К.	0,1667		
Ссылка = Магазин №23	0,1667		
Ссылка = ООО "БыстраяПицца"	0,1667		
Ссылка = ООО "Силай"	0,1667		
Ссылка = ИП Добров К.Л.	0,1667		
Ссылка = ООО "Лидер"		0,5000	
Ссылка = ООО "Василиса"		0,5000	
Ссылка = ООО "ДубМебель"			1,0000
КоличествоРозничныхТочек	1,3333	11,0000	1,0000
КоличествоАвтомобилей	1,1667	3,5000	10,0000
ВремяРаботыОрганизации = Меньше года	0,5000		
ВремяРаботыОрганизации = От года до трех	0,1667		
ВремяРаботыОрганизации = Свыше десяти лет	0,3333		
ВремяРаботыОрганизации = От трех до десяти лет		1,0000	1,0000
ВремяЗаключенияДоговора = Меньше года	0,6667	0,5000	
ВремяЗаклученияДоговора = От трех до десяти лет	0,1667		
ВремяЗаклученияДоговора = Свыше десяти лет	0,1667		
ВремяЗаклученияДоговора = От года до трех		0,5000	1,0000
ВидДоговора = Дилер	0,5000		
ВидДоговора = Постоянный партнер	0,5000	0,5000	
ВидДоговора = Дистрибьютор		0,5000	1,0000
ПрекращениеОтношений = Несоблюдение договора	0,1667		
ПрекращениеОтношений = Прекращение контрагентом	0,1667	1,0000	1,0000
ПрекращениеОтношений = Не прекращены	0,6667		

Расстояния между кластерами

Кластеры	1	2	3
1		8,7184	15,3968
2	8,7184		11,5370
3	15,3968	11,5370	

Рис. 3.4. Результат кластерного анализа

Введем необходимые обозначения: Во всех приведенных ниже формулах:

- X_i , Y_i – значения атрибутов двух объектов (между которыми определяется расстояние);
- W_i – весовой коэффициент атрибута (устанавливается в колонке анализа);

- i – номер атрибута, от 1 до n ;
- n – число атрибутов.

Евклидова метрика: расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sqrt{\sum (W_i * (X_i - Y_i)^2)}$$

Предположим, что объекты характеризуются одним свойством, которое у одного объекта имеет значение 9, у другого – 5. Весовой коэффициент данного атрибута равен единице. Расстояние между объектами будет равно:

$$\sqrt{\sum (1_1 * (9_1 - 5_1)^2)} = \sqrt{\sum (1 * (4)^2)} = 4$$

В метрике «*Евклидова метрика в квадрате*» расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sum (W_i * |X_i - Y_i|^2),$$

Предположим, что объекты характеризуются одним свойством, которое у одного объекта имеет значение 5, у другого – 3. Весовой коэффициент данного атрибута равен двум. Расстояние между объектами будет равно:

$$\sum (2_1 * (5_1 - 3_1)^2) = \sum (2 * 1 * (2)^2) = 8$$

В *Метрике города* расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \sum (W_i * |X_i - Y_i|),$$

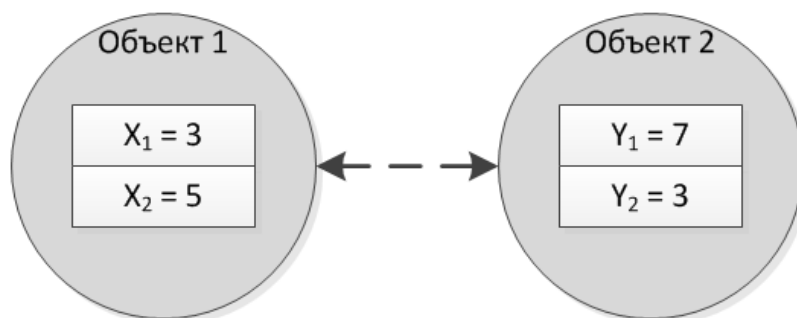


Рис. 3.5. Характеристики объектов

Предположим, что объекты характеризуются двумя атрибутами, которые имеют значения 3 и 5, 7 и 3. Вес первого равен 2, вес второго равен 1 (рис. 3.5).

Рассчитаем расстояние для объектов, представленных на рис. 3.5.

$$(W_1 * |X_1 - Y_1|) + (W_2 * |X_2 - Y_2|) = (2 * |3 - 7|) + (1 * |5 - 3|) = 10$$

В *Метрике доминирования* расстояние между двумя объектами вычисляется по формуле:

$$\text{Расстояние} = \text{MAX} (W_i * |X_i - Y_i|),$$

Рассчитаем расстояние для объектов, представленных на рис. 3.5.

$$\text{MAX} (W_1 * |X_1 - Y_1|, W_2 * |X_2 - Y_2|) = \text{MAX} (2 * |3 - 7|, 1 * |5 - 3|) = 8$$

Методы кластеризации

Вариант метода кластеризации определяет, исходя из каких принципов объект соотносится к той или иной группе, по какому алгоритму производится формирование кластеров.

Можно сказать, что целью любого алгоритма кластеризации является:

- минимизация изменчивости внутри кластеров,
- максимизация изменчивости между кластерами.

Различия между методами будем рассматривать на объектах, представленных на рисунке (см. рис. 3.6).

Будем считать, что объекты образуют две группы. Первая состоит из объектов 1, 2 и 3. Вторая группа состоит из объектов 4, 5 и 6.

Ближняя связь

Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до ближайшего объекта минимально.

В рассматриваемом примере объект 7 будет включен в группу, в которой находится объект 4. Самыми близкими объектами двух групп являются объекты 4 и 3. Расстояние до объекта 4 минимально.

Дальняя связь

Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до наиболее дальнего объекта минимально.

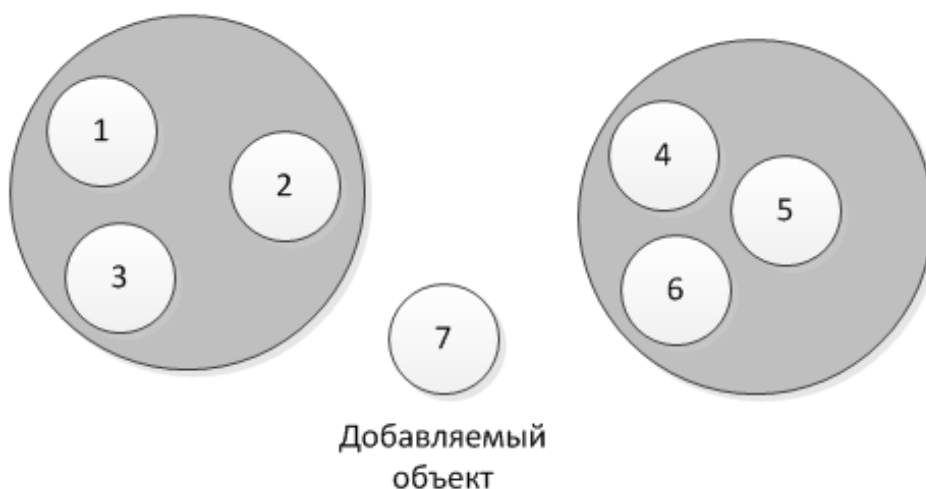


Рис. 3.6. Группы объектов

В рассматриваемом примере (рис. 3.7) объект 7 будет включен в группу, в которой находится объект 5. Самыми дальними объектами двух групп являются объекты 1 и 5. Расстояние до объекта 5 меньше.

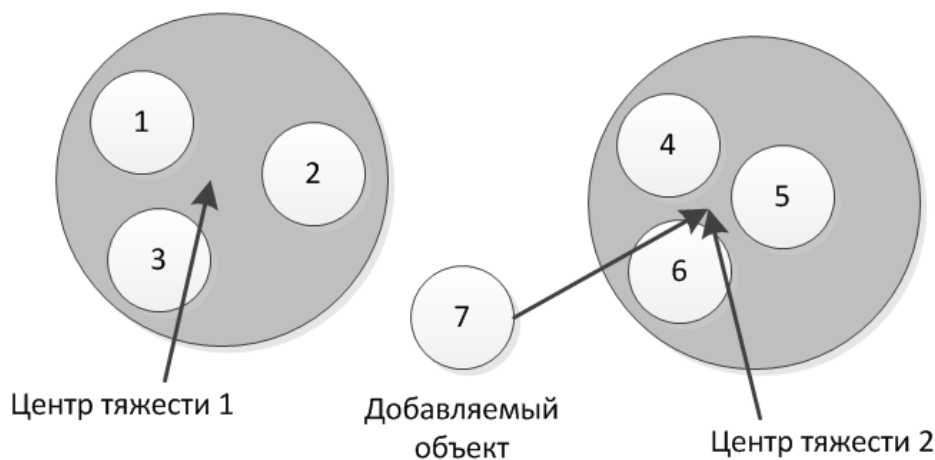


Рис. 3.7. Включение объекта в группу объектов

Центр тяжести

Метод кластеризации, в котором объект присоединяется к той группе, для которой расстояние до центра тяжести минимально.

В примере, рассмотренном на рисунке, объект 7 добавится в группу, содержащую объекты 4, 5 и 6. Расстояние до центра тяжести (некоего мифического объекта со средними значениями атрибутов) минимально.

k-средних

В данном методе выбираются объекты, находящиеся первыми в выборке. Они считаются центрами кластеров. Далее выбирается следующий объект и, в соответствии с расстоянием до центров кластеров, относится к тому или иному кластеру. Центр кластера, к которому был добавлен объект, пересчитывается.

Процедура повторяется до полного перебора всех объектов. Далее опять производится новая выборка объектов (начиная с первого). Процедура повторяется до тех пор, пока изменяются центры кластеров.

Предположим, что произвольно выбраны в качестве центров кластеров объекты 1 и 2 (рис. 3.8). Объект 3 добавляется к кластеру, центром которого является объект 1. Центр первого кластера перерасчитывается (он находится между объектом 1 и 3). Объект 4 добавляется ко второму кластеру (его центр также перерасчитывается).

После перебора всех анализируемых объектов к первому кластеру относятся объекты 1 и 3, ко второму – остальные объекты (его центр предположительно находится в центре треугольника из объектов 4, 7, 6).

Далее опять производится выборка объектов и распределение их по кластерам (относительно постоянно рассчитываемых центров кластеров).

Где-то на третьей выборке объектов, скорее всего, объект 2, который изначально был центром второго кластера, станет относиться к первому кластеру.

В конце алгоритма к первому кластеру будут относиться объекты 1, 2, 3. Ко второму – объекты 4, 5, 6, 7.

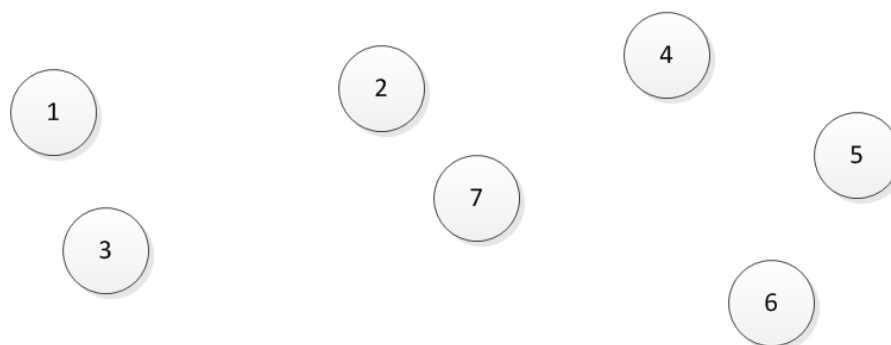


Рис. 3.8. Пример расположения объектов

Вывод данных в дендрограмму

При выводе данных кластерного анализа, если используется алгоритм, отличный от алгоритма k-средних, результаты кластерного анализа выводятся в виде дендрограммы (алгоритм анализа должен предусматривать вывод распределения анализируемых объектов по кластерам), рис. 3.9.

Дендрограмма связей

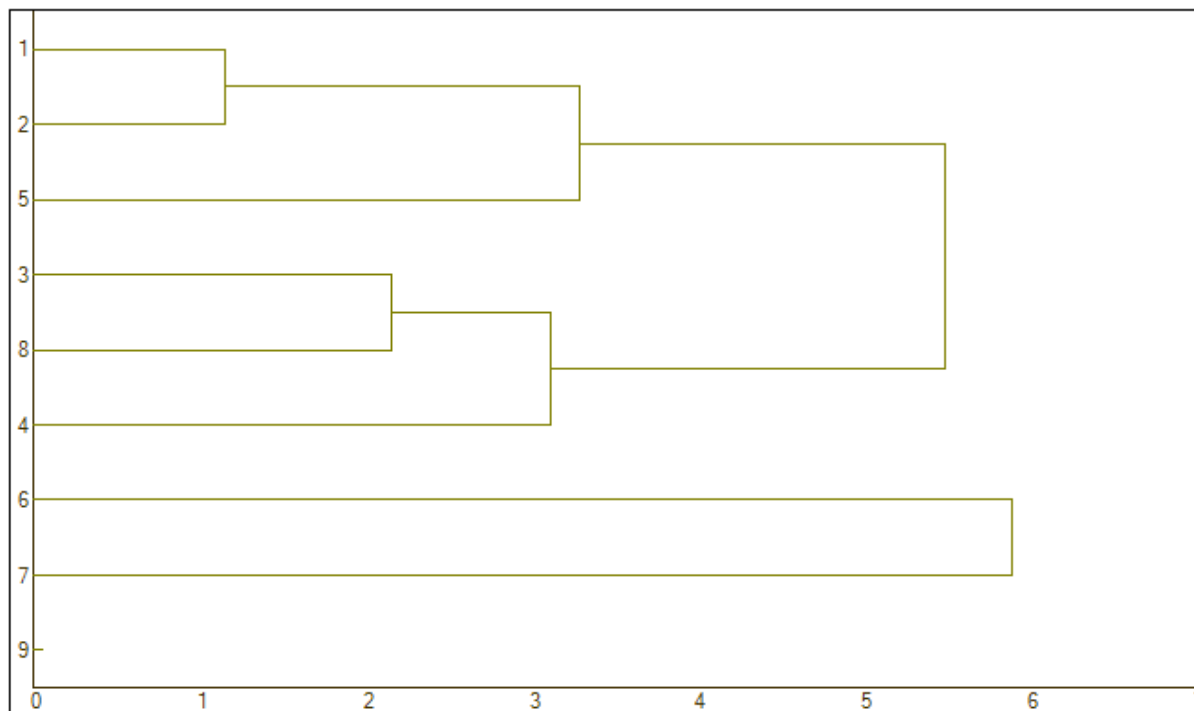


Рис. 3.9 Дендрограмма связей

Таким образом, кластерный анализ проводится чтобы выделить сегменты или кластеры из множества объектов одной природы. Объекты распределяются по группам так, чтобы внутригрупповые отличия были минимальными, а межгрупповые — максимальными. Методы кластеризации позволяют перейти

от пообъектного к групповому представлению совокупности произвольных объектов, что существенно упрощает оперирование ими.

Вот возможные сценарии применения кластеризации на практике:

– Сегментация клиентов по определенной совокупности параметров позволяет выделить среди них устойчивые группы, имеющие сходные покупательские предпочтения, уровни продаж и платежеспособности, что существенно упрощает управление взаимоотношениями с клиентами.

– При классификации товаров часто используются достаточно условные принципы классификации. Выделение сегментов на основе группы формальных критериев позволяет определить действительно однородные группы товаров. В условиях широкой и довольно разнородной номенклатуры товаров управление ассортиментом на уровне сегментов, по сравнению с управлением на уровне номенклатуры, существенно повышает эффективность продвижения, ценообразования, мерчендайзинга, управления цепочками поставок.

– Сегментация менеджеров позволяет эффективнее спланировать организационные изменения, улучшить мотивационные схемы, скорректировать требования к нанимаемому персоналу, что в конечном счете позволяет повысить управляемость компании и стабильность бизнеса в целом.

Сходство и различие между объектами определяется «расстоянием» между ними в пространстве факторов. Способ измерения расстояния определяется метрикой, которая указывает принцип определения сходства-различия между объектами выборки. МАДП содержит сейчас несколько метрик.

Способы формирования кластеров на основе информации о расстоянии между кластеризуемыми объектами определяются методом кластеризации. Любой из реализованных в платформе методов кластеризации предполагает явное указание количества искомых кластеров. Для атрибутов объектов можно вводить веса, что позволяет расставлять приоритеты между ними.

Результатами анализа с помощью кластеризации являются:

- центры кластеров, представляющие собой совокупность усредненных значений входных колонок в каждом кластере;
- таблица межкластерных расстояний (между центрами кластеров), определяющих степень различия между ними;
- значения прогнозных колонок для каждого кластера;
- рейтинг факторов и дерево условий, определивших распределение объектов на кластеры.

Алгоритмы кластеризации позволяют не только провести кластерный анализ объектов на множестве заданных атрибутов, но и спрогнозировать значение одного или нескольких из них для актуальной выборки на основании отнесения объектов этой выборки к тому или иному кластеру.

Тема 4. Механизм запросов, табличный способ доступа к данным. Язык запросов в «1С:Предприятии»: роль языка запросов, табличная модель представления данных, структура запроса, выражения в языке запросов

Роль языка запросов. Табличная модель представления данных. Какие прикладные объекты представляются таблицами языка запросов. Структура запроса (описание запроса). Указание списка полей выборки. Псевдонимы полей и таблиц. Выражения в языке запросов: выражения в списке полей выборки; условные выражения в языке запросов, оператор ВЫБОР; фильтрация результатов запроса, операция ГДЕ; логические выражения в языке запросов; параметры в языке запросов; встроенные функции в выражениях языка запросов; задание в запросе значений предопределенных данных конфигурации, функция ЗНАЧЕНИЕ().

В системе «1С:Предприятие» получить актуальные данные можно двумя действенными методами:

- объектный вариант получения информации;
- табличный способ извлечения данных.

Объектная модель используется тогда, когда цель – получить простой набор данных простым набором кода. Язык запросов в 1С работает для всех случаев, в этом его универсальность. Но неграмотно созданный запрос ухудшает быстродействие и не всегда даёт возможность получить корректные данные.

Когда эффективен табличный способ извлечения данных? – Когда требуется не только извлечь данные, но и выполнить группировку полей, расчет минимального, максимального, среднего значения, расчет итогов, сортировку строк, и т.п., то эффективно использование запросов. Следует помнить, что в системе «1С:Предприятие» запросы данные из ИБ только читают, не изменяют.

Запрос выполняется в соответствии с заданными инструкциями — текстом запроса. Текст запроса составляется в соответствии с синтаксисом и правилами языка запросов. Язык запросов «1С:Предприятие 8» основан на базе стандартного SQL, но имеет некоторые отличия и расширения, а также русскоязычный синтаксис.

Работа с запросом во встроенном языке включает нескольких последовательных этапов:

- создание объекта Запрос и установка текста запроса;
- установка параметров запроса;
- выполнение запроса и получение результата;
- обход результата запроса и обработка полученных данных.

Для формирования и выполнения запросов к таблицам базы данных в системе используется специальный объект Запрос. Запрос незаменим в случае необходимости получить сложную выборку данных, где данные должны быть определенным образом сгруппированы, как-то отсортированы и т.п. Одним из классических примеров его применения может служить сводка по состоянию регистра учета на определенный момент времени. Кроме того, механизм

запросов позволяет легко получать информацию в различных временных разрезах.

Запрос – это специальный объект в 1С 8.3, который используется для формирования и выполнения запросов к таблицам базы данных в системе. Для выполнения запроса необходимо составить текст запроса, в котором описывается какие таблицы будут использоваться в качестве источников данных запроса, какие нужно выбрать поля, какие применить сортировки и группировки и т.д. [1].

Понятия, тесно связанные с запросами:

- 1) Источники данных для запроса (таблицы системы, состав полей таблиц);
- 2) Структура запроса, язык написания запроса;
- 3) Порядок работы с запросом, получение данных, обход полученного результата.

Таблицы и поля базы данных

Разработчик прикладного решения не имеет возможности работать с таблицами базы данных (физическими таблицами). Язык запросов работает по «логическим таблицам». Это связано с тем, что в прикладном решении могут использоваться разные СУБД, имеющие свою специфику, а текст запроса должен быть универсальным и одинаково работать на любой используемой СУБД. Поэтому при выполнении запроса платформа автоматически транслирует текст запроса в набор инструкций, которые «понимает» конкретная СУБД.

Кроме того, физические таблицы и поля в них имеют техногенные имена, из которых непонятно, что именно хранится в данном поле.

Поэтому с помощью запросов мы обращаемся к данным не напрямую, а через специальную «прослойку» в виде таблиц языка запросов. Этот процесс можно представить на следующей схеме (рис. 4.1).

Логические таблицы в 1С:Предприятие» принято делить на два основных класса: реальные и виртуальные. Упрощенно их назначение можно сформулировать так:

Реальные таблицы хранятся в БД. В запросе, где они используются, можно конструировать вычисляемые поля, значения которых вычисляются как функция нескольких разных полей. В рамках класса реальных таблиц, отдельный подкласс таблиц составляют объектные таблицы (они хранят состояния объектов системы, имеющих объектную природу и в них присутствует поле «Ссылка»).

Виртуальные таблицы. Они в БД не хранятся. При обращении к информации виртуальных таблиц система автоматически собирает информацию реальных таблиц для выполнения запроса [1]. Виртуальная таблица может быть параметризована.

В свою очередь любая таблица состоит из набора полей (рис. 4.2). В качестве поля таблицы может выступать:

а) «Обычное поле». Содержит некое значение или «Null». В одной записи в обычном поле хранится единственное значение.

б) Вложенная таблица. Ей соответствует значение типа «РезультатЗапроса» с заранее заданным набором колонок.

Таким образом, все таблицы, к которым можно обратиться с помощью языка запросов, являются придуманными, воображаемыми, а реальные таблицы можно считать в большей степени соответствующими реальным физическим таблицам СУБД.

Структура запроса (описание запроса)

Написание запроса производится на специальном языке запросов, поддерживаемом платформой «1С:Предприятие». Текст запроса имеет определенную структуру [1]. Это инструкция, в соответствии с которой должен быть выполнен запрос (источники данных запроса, поля таблиц, которые требуется обрабатывать в запросе, правила группировки, сортировки результатов и т. д.).

Все ключевые слова языка запросов системы «1С:Предприятие» имеют два варианта написания: на русском и английском языках.

Возможная структура запроса следующая (см. ниже). В минимальном варианте текст запроса может содержать только конструкция «Выбрать <Поля выборки>».

```

ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] [ПЕРВЫЕ <Количество>]
    <Список полей выборки>
ИЗ <Список источников>]
ГДЕ <Условие отбора>]
СГРУППИРОВАТЬ ПО <Поля группировки>]
ИМЕЮЩИЕ <Условие отбора>]
    
```

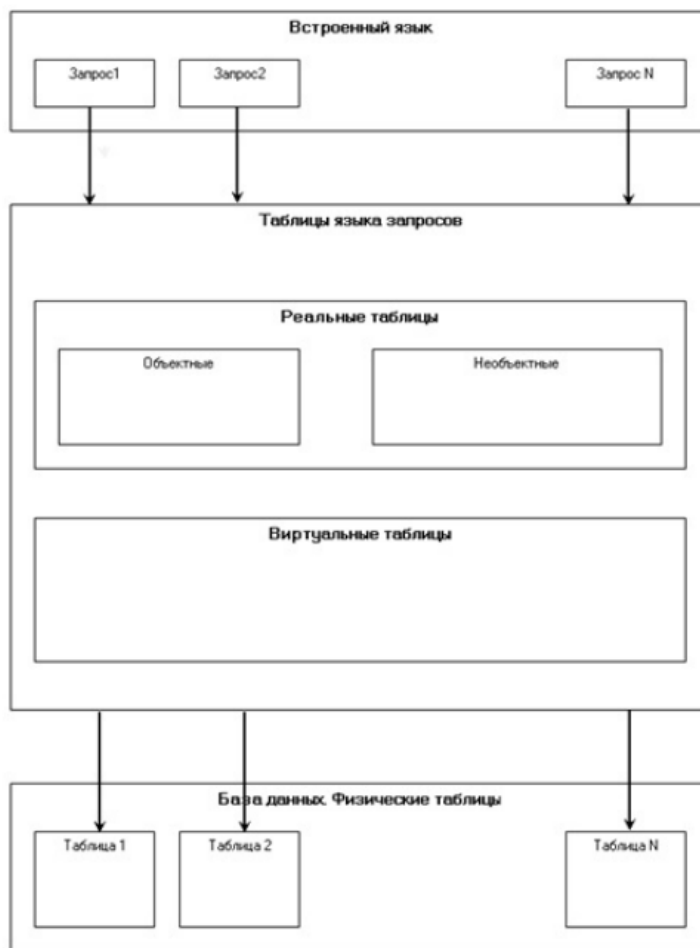


Рис. 4.1. Доступ к данным в «1С:Предприятии»

Дата	Продавец	Товары
01.09.2003 15:00:10	Продавцов П.П.	
08.09.2003 17:10:16	Продавайлов В.В.	

Товар	Количество
Ручка	10
Карандаш	50

Товар	Количество
Метла	5

Рис. 4.2. Вложенное поле

[ДЛЯ ИЗМЕНЕНИЯ [<Список таблиц верхнего уровня>]]
 [<Объединение запросов>]
 [<Упорядочивание результатов>]
 [АВТОУПОРЯДОЧИВАНИЕ]
 [<Описание итогов>]

<Описание запроса> – это единственная обязательная секция в тексте запроса, и во многих случаях достаточно указать только ее. В секции определяются источники данных запроса, поля выборки, группировки и т. д.

Указание списка полей выборки. Псевдонимы полей и таблиц.

После обязательного ключевого слова «Выбрать» следует описание полей выборки. Описание полей разделяются запятыми. После последнего описанного поля запятая не ставится [1].

Каждое <Поле выборки> состоит из описания поля выборки и необязательного псевдонима поля.

Вместо перечисления полей в списке выборки можно указать звездочку «*». Это будет означать, что результат запроса должен содержать все поля, которые есть в исходных таблицах - источниках данных запроса, описанных в списке источников.

Псевдонимы полей. Каждому полю выборки может быть назначен **псевдоним**. В дальнейшем <Псевдоним поля> может использоваться для более удобного обращения к данному полю. Если полю выборки назначить псевдоним, то в дальнейшем к этому полю можно будет обращаться, используя его псевдоним, в предложениях УПОРЯДОЧИТЬ ПО и ИТОГИ, а также при работе с результатом запроса. Такое обращение может быть более удобным и наглядным, а в некоторых случаях единственно возможным. Ключевое слово КАК может предшествовать псевдониму поля. Это слово можно не указывать вообще, но если оно указано, повышается наглядность и удобочитаемость текста запроса [1].

Источник запроса (таблица) указывается после ключевого слова «Из». В общем случае источников может быть несколько. Описание источников разделяется запятыми, после последнего источника запятая не ставится.

ВЫБРАТЬ

Ссылка КАК Документ,

Ответственный КАК МатериальноОтветственноеЛицо

ИЗ Документ.АвансовыйОтчет

УПОРЯДОЧИТЬ ПО МатериальноОтветственноеЛицо

Псевдонимы источников данных. Если источнику данных назначить псевдоним, то в дальнейшем к этому источнику можно будет обращаться, используя этот псевдоним (и уже нельзя будет обращаться через указание имени таблицы). Такое обращение может быть более удобным и наглядным, а в некоторых случаях единственно возможным [1].

Псевдоним задается в соответствии с правилами назначения идентификаторов переменных. Псевдонимы в запросе не могут совпадать.

Ключевое слово КАК может предшествовать псевдониму источника. Это слово можно не указывать вообще, но если оно указано, повышается наглядность и удобочитаемость текста запроса.

Назначение псевдонимов источникам само по себе никак не влияет на выборку данных в запросе.

Пример [1]:

```
// Исходной таблице-источнику
// назначен псевдоним Товар
Справочник.Номенклатура
Выбрать
    Товар.Наименование,
    Товар.Родитель
Из
    Справочник.Номенклатура КАК Товар
```

Оператор Выбор. При работе с языком запросов 1С, иногда возникает ситуация, когда требуется проверить выражение, является ли оно ЛОЖЬ или Истина. В случае положительного результата присвоить ему значение выражения для замены [1].

Оператор ВЫБОР имеет конструкцию [1]:

```
Код
ВЫБОР
КОГДА <Выражение> ТОГДА <Выражение>
ИНАЧЕ <Выражение>
КОНЕЦ
```

Причем ветку КОГДА <Выражение> ТОГДА <Выражение> можно использовать несколько раз.

На закладке «Дополнительно» можно задать ряд дополнительных опций (рис. 4.2).

Рассмотрим их назначение [1].

1. Первые (n) – позволяет выбрать заданное число записей, первых в запросе, с учетом заданного порядка записей. В тексте запроса сгенерит ключевое слово **ВЫБРАТЬ ПЕРВЫЕ N**.

Конструкция **ПЕРВЫЕ <Количество>** позволяет указать какое предельное количество строк отобразить в результат запроса. В соответствии с правилами упорядочивания результатов запроса будут отобраны самые первые строки. <Количество> – целое число.

Пример:

```
// Необходимо показать 3 товара с наиболее высокой отпускной
// ценой. Сортируем в порядке убывания цены товара.
Выбрать Первые 3
Справочник.Номенклатура.Наименование,
Справочник.Номенклатура.ОтпускнаяЦена
Упорядочить По
Справочник.Номенклатура.ОтпускнаяЦена Убыв
```

2. Опция «Без повторяющихся» оставит в детальных записях выборки только уникальные записи. В тексте запроса сгенерит ключевое слово **ВЫБРАТЬ РАЗЛИЧНЫЕ**

Иногда нам нужно, чтобы одинаковые строки в отчете не повторялись. Например, в тексте запроса ниже мы хотим получить список всех контрагентов, которым осуществлялась документом Расходная Накладная отгрузка товаров. Естественно, что контрагент мог осуществлять покупку не один раз, чтобы исключить в результате запроса дублирование данных, можно использовать конструкцию **РАЗЛИЧНЫЕ**. Текст запроса, который сформирует неповторяющийся перечень контрагентов, кому мы отгружали товары:

Выбрать Различные
Документ.РасходнаяНакладная.Контрагент

3. Разрешенные – необходимо для настройки ограничений прав доступа; будут показаны только разрешенные для данных полномочий записи (если опцию не ставить, то при ограниченных правах запрос может вернуть ошибку и не выполниться). В тексте запроса опция сформирует ключевое слово **ВЫБРАТЬ РАЗРЕШЕННЫЕ**.

Это означает, что запрос выберет только те записи, на которые у текущего пользователя есть права. Если данное слово не указать, то запрос отработает с ошибкой, когда он выберет 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 (см. здесь). Права доступа на саму таблицу не учитываются при использовании ключевого слова **РАЗРЕШЕННЫЕ**. Например, 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**.

4. Тип запроса - выборка данных либо создание временной таблицы. Опция временной таблицы позволяет задать ее имя; к данной таблице впоследствии можно будет обращаться как промежуточному источнику данных редактируемого запроса. В тексте запроса в этом случае появится ключевое слово **ПОМЕСТИТЬ**.

ПОМЕСТИТЬ <имя временной таблицы>

В пакетном запросе создает временную таблицу с заданным именем, которая будет существовать до окончания исполнения всего пакета запроса или до исполнения в пакете запроса, уничтожающего данную временную таблицу с помощью конструкции **УНИЧТОЖИТЬ**.

5. Блокировать полученные данные для последующего изменения - предполагается, что пока идет чтение из таблиц-источников, они блокируются на чтение в иных сессиях. В тексте запроса в случае включения опции появится ключевое слово **ДЛЯ ИЗМЕНЕНИЯ**. При этом можно указать, какие именно таблицы из числа задействованных в запросе блокируются; если этого не сделать, то заблокируются все.

Конструкция **ДЛЯ ИЗМЕНЕНИЯ** используется, когда считываемые в транзакции данные надо заблокировать от чтения в других сессиях. Для файлового варианта блокируются указанные таблицы, а для клиент-серверного варианта - только выбранные записи. Блокировка снимается после завершения

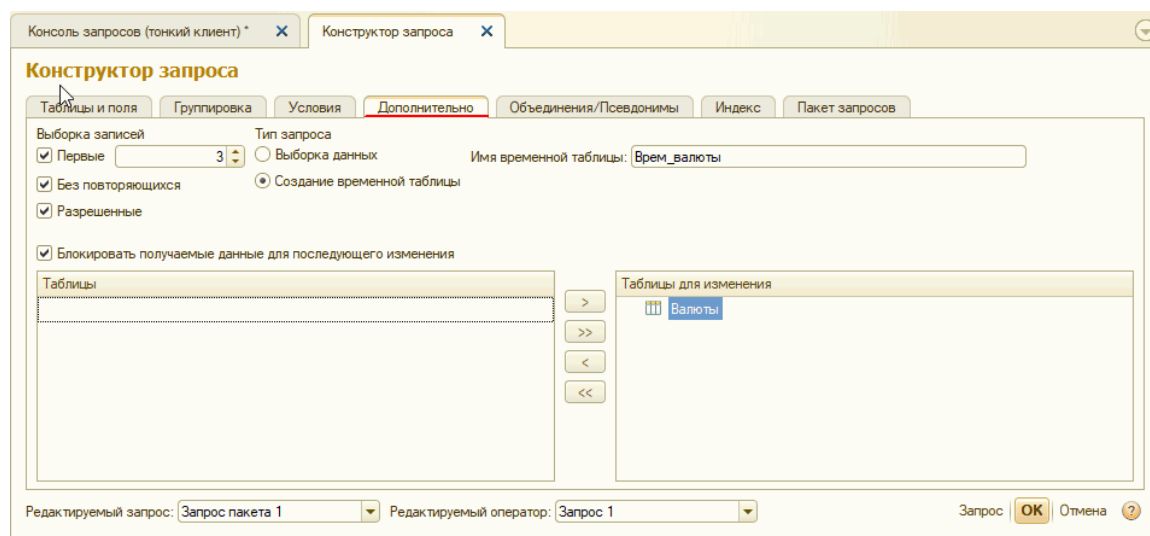


Рис. 4.2. Вкладка «Дополнительно»

транзакции.

Фильтрация результатов запроса. Предложение **ГДЕ** <Условие отбора> позволяет задать условие отбора данных из исходных таблиц - источников запроса; в запросе будут обрабатываться только те записи, для которых данное условие оказывается истинным.

ВЫБРАТЬ

Дисциплина .Наименование,
 Дисциплина .ОбщийОбъемВЧасах,
 Дисциплина .ЧасовЛекций,
 Дисциплина .ЧасовПрактическихЗанятий,
 Дисциплина .ЧасовЛабораторныхЗанятий,
 Дисциплина .ЧасовСамостоятельнойРаботы,
 Дисциплина .ВидОтчетности

ИЗ

Справочник .Дисциплина КАК Дисциплина

ГДЕ Дисциплина .ВидОтчетности =
 Значение (Перечисление .ВидыОтчетности .Зачет)

Условие отбора может быть простым логическое выражением или сложным, в котором логическими операторами **И**, **Или**, **Не** связано несколько простых логических выражений.

При выборке данных из таблицы, имеющей табличные части, условия на записи табличной части действуют как на состав записей, выбираемых из таблицы, так и на состав записей табличных частей.

В общем виде: выборка записей, содержащих табличные части, выполняется в следующем порядке:

- К каждой таблице из списка источников левым внешним соединением присоединяются ее табличные части.
- Выполняются все соединения, определенные в списке источников.
- Отбираются записи, удовлетворяющие условию из раздела ГДЕ.
- Выполняется группировка записей по полям таблиц из списка источников, включая поля ссылок, с помещением записей табличных частей в качестве значений соответствующих полей записей таблиц из списка выборки.

Условие содержит выражение отбора значений реквизитов, которое определяет включение и использование кортежа в результирующей выборке. Выражение условия может использовать любые реквизиты из любых источников определенных разделе ИЗ запроса.

- `<псевдоним>.<реквизит> {=,<, >, <>, В, НЕ В} <критерий> [{И, ИЛИ} ...]` — базовая форма выражения условия
 - `<псевдоним>.<реквизит> В (<значение1>,<значение2>,...)` — для условия вхождения в множество допустимо перечисление произвольного набора значений
 - `<псевдоним>.<реквизит> НЕ В (...)` — для условия отсутствия в множестве
 - `<псевдоним>.<реквизит> В (ВЫБРАТЬ ...)` — для условия В допустимо выражение вложенного запроса любой сложности, который возвращает только одну колонку
 - `<псевдоним>.<реквизит>.<реквизит>...` — разыменование допустимо, но ведет к неявному объединению, поэтому желательно избегать
 - `<псевдоним1>.<реквизит1> {=,<, >, <>, В} {<псевдоним2>.<реквизит2>, <константа>, &<параметр запроса>,(<выражение вложенного запроса>)} [{И, ИЛИ} ...]` — корректные формы выражения условия
 - МЕЖДУ —
 - `<псевдоним>.<реквизит> ПОДОБНО <строка шаблона> [СПЕЦСИМВОЛ <строка>]` — сравнивает по шаблону, где
 - % — допускает любую последовательность произвольных символов
 - _ — любой одиночный символ
 - [...] — один произвольный символ из перечисленных внутри [АЯОЁУЮЭЕ] или заданных диапазоном [А-F];
 - [^...] — то же самое, что и предыдущее, но наоборот. Знак «^» означает отрицание.
 - СПЕЦСИМВОЛ <строка> — определяет префиксный символ (рекомендуется: #,\,/,\~) после которого в шаблоне символы %, _, [,] означают просто символ
 - `<реквизит дата> = ДАТАВРЕМЯ(<реквизит года>, <реквизит месяца>, <реквизит числа>, <реквизит часа>, <реквизит минуты>, <реквизит секунды>)` —
 - `<реквизит> ЕСТЬ NULL` — выражение равно Истина, если значения NULL

- НЕ (<реквизит> ЕСТЬ NULL) — выражение равно Истина, если значение не NULL (**NULL — это не пустое значение**)
 - <псевдоним>.<реквизит> = ЗНАЧЕНИЕ (Справочник.<справочник>.ПустаяСсылка) — сравнение с пустым значением
 - <псевдоним>.<реквизит> = ЗНАЧЕНИЕ(Справочник.<справочник>.<предопределенное значение>) — сравнение с предопределенным значением
 - <псевдоним>.<реквизит> = ЗНАЧЕНИЕ(Перечисление.<имя перечисления>.<значение перечисления>)
 - <псевдоним>.<табличная часть>.Ссылка — значение ссылки в табличной части эквивалентно значению <псевдоним>.Ссылка
 - <ссылочный реквизит>.**Ссылка** = &<параметр условия> — неправильное избыточное использование реквизита Ссылка вызывает ненужное соединение с таблицей ссылки, правильным будет не использовать «.Ссылка»
 - <ссылочный реквизит> ССЫЛКА <тип> — для проверки соответствия типа ссылки, например: Учет.Регистратор ССЫЛКА Документ.Накладная
 - ТИПЗНАЧЕНИЯ(<ссылочный реквизит>) = ТИП(Справочник.<имя справочника>)
 - <справочник>.Родитель = &Группа — условие отбора кортежей непосредственно входящих в группу иерархического справочника
 - <справочник>.Ссылка В ИЕРАРХИИ(&Группа) — условие отбора всех входящих в группу и подгруппы
 - <справочник>.Родитель = ИЕРАРХИИ(&Группа) — условие отбора всех входящих в группу и подгруппы
 - <справочник>.Родитель.Родитель.Родитель... — получение цепочки родителей иерархического элемента нативными средствами запроса не предусмотрено
 - <выражение> В (ВЫБРАТЬ <реквизит> ИЗ <Источники> ГДЕ <Условие>) — для проверки вхождения значения выражения в набор из отдельной выборки
 - (<выражение 1>, ..., <выражение N>) В (ВЫБРАТЬ <реквизит 1>, ..., <реквизит N> ИЗ <Источники> ГДЕ <Условие>...) — для проверки вхождения группы значений в набор групп [/its.1c.ru](https://its.1c.ru)
 - если при **ЛЕВОЕ СОЕДИНЕНИЕ** задано условие к реквизиту правой таблицы (или наоборот), то фактически выполнится **ВНУТРЕННЕЕ СОЕДИНЕНИЕ**

Встроенные функции в выражениях языка запросов.

Могут использоваться в выражениях в списке полей выборки и в условии отбора в предложении **ГДЕ**:

```

ПОДСТРОКА (<Выражение>, <Значение>, <Значение>) |
ГОД (<Выражение>) |
КВАРТАЛ (<Выражение>) |
МЕСЯЦ (<Выражение>) |
ДЕНЬГОДА (<Выражение>) |
ДЕНЬ (<Выражение>) |
НЕДЕЛЯ (<Выражение>) |
ДЕНЬНЕДЕЛИ (<Выражение>) |
ЧАС (<Выражение>) |
МИНУТА (<Выражение>) |
СЕКUNДА (<Выражение>) |
НАЧАЛОПЕРИОДА (<Выражение>, Минута | Час | День | Неделя |
Месяц | Квартал | Год | Декада | Полугодие) |
КОНЕЦПЕРИОДА (<Выражение>, Минута | Час | День | Неделя |
Месяц | Квартал | Год | Декада | Полугодие) |
ДОВАВИТЬКДАТЕ (<Выражение>, Минута | Час | День | Неделя | |
Месяц | Квартал | Год | Декада | Полугодие, <Выражение>) |
РАЗНОСТЬДАТ (<Выражение>, <Выражение>, Секунда | Минута | Час
| День | Месяц | Квартал | Год) |
ТИПЗНАЧЕНИЯ (<Выражение>) |
ПРЕДСТАВЛЕНИЕ (<Выражение>) |
ЕСТЬNULL (<Выражение>, <Выражение>) |
АВТОНОМЕРЗАПИСИ ( )
СТРУППИРОВАНАПО (<Выражение>)

```

Литерал ЗНАЧЕНИЕ() для значений предопределенных данных конфигурации в запросе.

Литерал ЗНАЧЕНИЕ (<ПредставлениеЗначения>) используется в случаях, когда необходимо подставить в запрос, например, в условие, какие-то предопределенные данные конфигурации или значения системных перечислений.

Для системных перечислений представление значения имеет вид:

<ИмяСистемногоПеречисления>.<Значение>

Для предопределенных данных конфигурации представление значения имеет вид:

<ТипПредопределенногоЗначения>.<ИмяОбъектаМетаданных>.<Значение>

Тип предопределенного значения может быть:

- Справочник (Catalog);
- ПланВидовХарактеристик (ChartOfCharacteristicTypes);
- ПланСчетов (ChartOfAccounts);
- ПланВидовРасчета (ChartOfCalculationTypes);
- Перечисление (Enum).

ИмяОбъектаМетаданных – это имя объекта метаданных, заданное в конфигураторе.

Для определенных в конфигурации перечислений значение указывается как имя соответствующего объекта метаданных типа ЗначениеПеречисления. Для всех остальных типов предопределенных значений – как имя предопределенного элемента данных, как оно указано в конфигураторе, или ПустаяСсылка (EmptyRef) для указания пустой ссылки.

Тема 5. Механизм запросов, табличный способ доступа к данным. Язык запросов в «1С:Предприятии»: групповые операции в языке запросов, извлечение данных запросом, вложенные запросы

Групповые операции в языке запросов: группировка по значению поля и агрегатные функции; отбор по значению результата агрегатной функции, операция ИМЕЮЩИЕ. Упорядочивание записей результата запроса. Получение первых N записей результата запроса. Операция ПЕРВЫЕ. Получение неповторяющихся комбинаций значений полей результата запроса. Операция РАЗЛИЧНЫЕ. Извлечение данных запросом, хранящихся в таблицах разных объектов конфигурации. Декартово произведение (или "картезиан") множеств записей таблиц - источников данных. Наложение условия на картезиан. Операция ВНУТРЕННЕЕ СОЕДИНЕНИЕ таблиц - источников данных. Операция ВНУТРЕННЕЕ СОЕДИНЕНИЕ как современная форма записи условия формирования отношения. Операция ЛЕВОЕ СОЕДИНЕНИЕ языка запросов. Отличие операции ЛЕВОЕ СОЕДИНЕНИЕ от операции ВНУТРЕННЕЕ СОЕДИНЕНИЕ. Тип данных языка запросов NULL. Оператор ЕСТЬ NULL и функция ЕСТЬNULL(). Операция ПРАВОЕ СОЕДИНЕНИЕ языка запросов. Операция ПОЛНОЕ СОЕДИНЕНИЕ языка запросов. Операция разыменования поля ссылочного типа. Обращение к табличной части объекта как к таблице - источнику данных. Поле Ссылка таблицы, представляющей табличную часть. Вложенные запросы. Объединение результатов нескольких запросов.

Работа с запросами на встроенном языке системы «1С:Предприятие».

Группирующие наборы

Простая группировка (предложение **СГРУППИРОВАТЬ ПО**) позволяет вычислить заданные агрегатные функции для каждого набора данных, в котором находится уникальная комбинация значений для полей, перечисленных в предложении **СГРУППИРОВАТЬ ПО**.

Предложение **ИМЕЮЩИЕ <Условие отбора>** позволяет накладывать условия на значения агрегатных функций. В других конструкциях языка запросов, например, в предложении **ГДЕ**, указывать в условиях агрегатные функции нельзя [1].

```

ВЫБРАТЬ
    ПродажаТоваров.Номенклатура,
    СУММА (ПродажаТоваров.Сумма) КАК Сумма
ИЗ
    Документ.ПродажаТоваров.Товары КАК ПродажаТоваров
Сгруппировать По
    ПродажаТоваров.Номенклатура
ИМЕЮЩИЕ
    СУММА (ПродажаТоваров.Сумма) > 15000
  
```

При группировке результатов запроса и при подсчете итогов в запросах могут использоваться агрегатные функции, которые используются для обобщения значений указанного параметра. В языке запросов 1С определены следующие агрегатные функции:

```
СУММА (<Выражение> |
СРЕДНЕЕ (<Выражение> |
МИНИМУМ (<Выражение> |
МАКСИМУМ (<Выражение> |
КОЛИЧЕСТВО ( [РАЗЛИЧНЫЕ] <Выражение> | * )
```

Конструкция, о которой часто забывают, но она очень важна и полезна. Она позволяет указать отбор в виде агрегатной функции, этого нельзя сделать в конструкции ГДЕ.

Еще один пример использования конструкции ИМЕЮЩИЕ в запросе 1С:

ВЫБРАТЬ

```
ПоступлениеТоваровУслугТовары.Товар,
СУММА(ПоступлениеТоваровУслугТовары.Количество)
КАК Количество,
СУММА(ПоступлениеТоваровУслугТовары.Сумма) КАК
Сумма
ИЗ
```

```
Документ.ПоступлениеТоваровУслуг.Товары КАК
ПоступлениеТоваровУслугТовары
```

СГРУППИРОВАТЬ ПО

```
ПоступлениеТоваровУслугТовары.товар
```

ИМЕЮЩИЕ

```
СУММА(ПоступлениеТоваровУслугТовары.Количество) > 5
```

Так мы отберем количество товаров, которых поступило более 5 штук.

Упорядочивание результатов запроса

Предложение **УПОРЯДОЧИТЬ ПО** позволяет сортировать строки в результате запроса.

```
УПОРЯДОЧИТЬ ПО <Условия упорядочивания>
```

```
<Условия упорядочивания>
```

```
<Поле упорядочивания> [<Порядок>]
[, <Поле упорядочивания> [<Порядок>] [, ...]]
```

```
<Поле упорядочивания>
```

```
<Выражение> <Порядок> ВОЗР | УБЫВ | ИЕРАРХИЯ |
ИЕРАРХИЯ УБЫВ
```

В предложении **УПОРЯДОЧИТЬ ПО** через запятую может перечисляться несколько условий упорядочивания результата запроса. Выборка упорядочивается сначала по первому условию, потом по второму и т. д.

Упорядочивание может осуществляться в порядке возрастания или убывания значений, а для таблиц, для которых задано свойство иерархичности, - также и по иерархии. Порядок может задаваться для каждого поля независимо.

РАЗЛИЧНЫЕ — означает, что в результат не попадет дублирующих строк.

ПЕРВЫЕ n — в языке 1С 8.3, запрос отберет только n первых записей.

ЕСТЬNULL (ISNULL)

Достаточно полезная функция языка запросов 1С, которая проверяет значение в записи, и, если оно равно **NULL**, то позволяет заменить на своё значение. Чаще всего используется при получении виртуальных таблиц остатков и оборотов что бы скрыть **NULL** и поставить понятный 0 (ноль).

Пример:

ЕСТЬNULL(НалогиПредМесяца.ПримененнаяЛьготаФСС, 0)

Такая функция языка запросов 1С **ЕСТЬNULL**, при отсутствии значения вернет ноль, что позволит избежать ошибки.

ОБЪЕДИНИТЬ [ВСЕ]

Присоединяет к уже полученной выборке выборку следующего выражения.

<выражение> **ОБЪЕДИНИТЬ** <выражение> **ОБЪЕДИНИТЬ** <выражение>

... — число выражений объединения синтаксически не ограничивается
объединяемые выборки должны состоять из идентичных наборов колонок по типу и порядку

совпадение имен колонок не требуется

имена колонок объединенной выборки определяются первым выражением
[ВСЕ] определяет объединение повторяющихся записей без группировки,

иначе объединенная выборка повторяющихся записей содержать не будет

Нельзя объединить выборки, в которых содержится Строка неограниченной длины или Тип значения.

Разыменованное поле.

Разыменованное ссылочное поле – это операция, когда мы обращаемся к полям через точку.

Например, **Оплата.Контрагент.ВидКонтрагента**. В документе "Оплата" есть ссылочное поле «Контрагент». Оно ссылается на справочник "Контрагенты", в которой будет получено значение реквизита "ВидКонтрагента". Важно понимать, что при обращении к полям через точку платформа неявно создает подзапрос и соединяет эти таблицы.

Запрос:

ВЫБРАТЬ

Оплата.Ссылка,

Оплата.Контрагент,

Оплата.Контрагент.ВидКонтрагента

ИЗ

Документ.Оплата КАК Оплата

Можно представить в виде:

ВЫБРАТЬ

Оплата.Ссылка,

Оплата.Контрагент,

Контрагенты.ВидКонтрагента

ИЗ

Документ.Оплата КАК Оплата

ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Контрагенты КАК

Контрагенты

ПО Оплата.Контрагент = Контрагенты.Ссылка

Когда мы производим обращение через точку к ссылочному полю, имеющему составной тип, платформа пытается создать неявные соединения со всеми таблицами, которые входят в составной тип этого поля. Если ничего не предпринять, то запрос будет неоптимален. Если четко известно, какого типа поле, необходимо ограничивать такие поля по типу конструкции **ВЫРАЗИТЬ()**.

Например, имеется регистр накопления "Нераспределенные оплаты", где регистратором могут выступать несколько документов. В этом случае неверно получать значения реквизитов регистратора таким образом:

ВЫБРАТЬ

НераспределенныеОплаты.Регистратор.Дата,

.....

ИЗ

РегистрНакопления.НераспределенныеОплаты КАК

НераспределенныеОплаты

следует ограничить тип составного поля регистратор:

ВЫБРАТЬ**ВЫРАЗИТЬ(НераспределенныеОплаты.Регистратор КАК**

Документ.Оплата).Дата,

.....

ИЗ

РегистрНакопления.НераспределенныеОплаты КАК

НераспределенныеОплаты

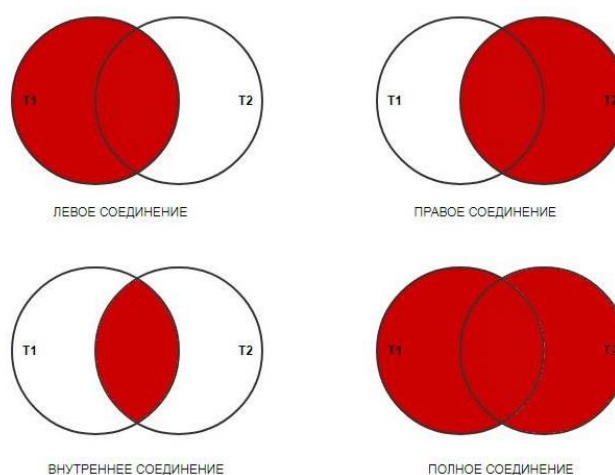
Когда в запросе используется более одного источника, для указания взаимосвязи таблиц могут использоваться соединения, которые бывают четырех видов (рис. 5.1):

- внутреннее соединение;
- левое внешнее соединение;
- правое внешнее соединение;
- полное внешнее соединение.

ЛЕВОЕ/ПРАВОЕ

СОЕДИНЕНИЕ. Суть **ЛЕВОГО** соединения заключается в том, что

полностью берется первая указанная таблица и к ней по условию связи привязывается вторая. Если записей, соответствующих первой таблице во второй не нашлось, то в качестве их значений подставляется NULL. Проще говоря, главной является первая указанная таблица и к её данным уже

**Рис. 5.1. Виды соединений**

подставляются данные второй таблицы (если они есть). В ПРАВОМ все в точности да наоборот.

ПОЛНОЕ СОЕДИНЕНИЕ

Данный вид соединения отличается от предыдущих тем, что в результате будут возвращены все записи как первой таблицы, так и второй. Если по заданному условию связи в первой или второй таблице не найдено записей, вместо них будет возвращено значение NULL.

ВНУТРЕННЕЕ СОЕДИНЕНИЕ

Отличием ВНУТРЕННЕГО соединения от ПОЛНОГО является то, что если хотя бы в одной из таблиц не найдена запись, то запрос не выведет ее вообще. В результате будут выбраны только те номенклатурные позиции из документа «Поступление товаров и услуг», для которых в регистре сведений «Цены номенклатуры» есть записи, если в предыдущем примере заменить «ПОЛНОЕ» на «ВНУТРЕННЕЕ».

Обращение к табличной части объекта как к таблице - источнику данных. Поле Ссылка таблицы, представляющей табличную часть.

Часто может понадобиться вывести все данные из табличной части определенного документа или элемента справочника или другого объекта конфигурации, имеющего табличную часть.

Для этого в языке запросов существует удобная возможность – обращаться к табличной части как к отдельной таблице. При этом в тексте запроса можно использовать все ранее изученные нами конструкции: ВЫБРАТЬ, ИЗ и т. п. Синтаксис обращения к таблице-источнику включает также имя табличной части – <Имя класса объектов>.<Имя объекта конфигурации>.<Имя табличной части>, например Документ.ЗаказТовара.Состав.

Следующий запрос выводит данные всех табличных частей из всех документов ЗаказТовара.

Листинг. Вывод данных всех табличных частей из всех документов «ЗаказТовара»:

ВЫБРАТЬ

СоставЗаказа.Товар,
СоставЗаказа.Количество,
СоставЗаказа.Сумма

ИЗ

Документ.ЗаказТовара.Состав КАК СоставЗаказа

Поля выборки запроса Товар, Количество, Сумма – это реквизиты табличной части документа, находящиеся в отдельной таблице базы данных, к которой мы обращаемся запросом.

Поскольку в нашем примере источником запроса является подчиненная таблица документа Документ.ЗаказТовара.Состав, то, чтобы получить данные из основной таблицы документа, нужно обращаться к полям таблицы через точку от поля табличной части Ссылка (листинг 1.40).

Листинг. Вывод данных из основной и подчиненной таблицы документов «ЗаказТовара»:

ВЫБРАТЬ

СоставЗаказа.Ссылка.Номер,
СоставЗаказа.Ссылка.Клиент,
СоставЗаказа.Товар,
СоставЗаказа.Количество,
СоставЗаказа.Сумма

Из

Документ.ЗаказТовара.Состав КАК СоставЗаказа

Примеры запросов:

I Работа в запросе с константами.

Значения всех констант содержатся в одной таблице, имя таблицы «Константы». На каждую константу в данной таблице определяется поле с именем поля, совпадающим с именем константы, задаваемым в конфигураторе.

1) Выбрать

Константы.Адрес,
Константы.ДатаУчрежденияКомпании

2) Выбрать

Адрес, ДатаБлокировки

Из

Константы

3) Выбрать

*

Из Константы

II Работа с вложенными таблицами в запросе

Есть справочник «Сотрудники» с табличной частью «Родственники». Табличной части соответствует вложенная таблица. Для обращения к этой вложенной таблице будет использоваться имя, состоящее из 3 частей (разделенных точками): ключевое слово «Справочник», «Имени справочника», «Имени табличной части».

Выбрать

*

Из

Справочник.Сотрудники.Родственники

III Конструкции «Различные», Первые №

К ключевому слову «Выбрать» могут использоваться «дополнения» для получения в результате запроса необходимого количества «первых» записей или чтобы в результате запроса были только «различные» записи.

Выбрать Первые № (количество записей)

Выбрать Различные

IV Фильтрация результатов запроса (отборы)

Для указания условия отбора используется конструкция «Где <Условие отбора>». Условие действует на момент получения данных запросом.

Примеры:

1) В условии может использоваться тот факт, что некоторые поля могут содержать значение логического типа.

Выбрать

Код, Наименование

Из

Справочник.Номенклатура

Где ЭтоГруппа

2) В условиях можно использовать стандартные логические операции

ЭтоГруппа=Ложь И (Не ПометкаУдаления)

3) Выбрать

Код, Наименование

Из

Справочник.Номенклатура

V Группировки в запросе

Данные в запросе могут быть сгруппированы с помощью агрегатных функций, указанных в качестве полей выборки.

Выбрать

Номенклатура Как Товар,

Сумма (Количество)

Из

Документ.ПоступлениеТовары.Товары

Сгруппировать по Номенклатура

При указании группировки псевдоним поля указывать нельзя.

В качестве агрегатных функций можно использовать:

Сумма(Выражение)

Среднее(Выражение)

Максимум(Выражение)

Минимум(Выражение)

Количество ([Различные] Выражение)

VI Условия на значения агрегатных функций описываются в отдельной конструкции «Имеющие».

Выбрать

Номенклатура Как Товар,

Сумма (Количество) ,

Сумма (Сумма)

Из

Документ.ПоступлениеТоваров.Товары

Сгруппировать По Номенклатура

Имеющие Сумма (Сумма) >1000

VII Указание нескольких источников и соединение таблиц

Не всегда данные, необходимые для выборки, находятся в одной таблице. Иногда приходится (явным или неявным образом) в качестве источников запроса указывать несколько таблиц.

Использованием соединений для взаимосвязи таблиц. Соединения бывают нескольких видов, рассмотрим их варианты на простом примере.

Допустим, есть две таблицы, содержащие информацию о самолетах (рейсах) и пассажирах.

Самолеты	
Наименование	Номер
ИЛ86	АА1
Ту 154	АА2
ИЛ 62	АА3

Пассажиры	
НамерСамолета	ФИО
АА1	Петров
АА2	Вольский
АА4	Третьяков
АА1	Чемисов

Условием соединения будет

Самолеты.Номер=Пассажиры.НомерСамолета

В результате запроса должны попасть данные

Наименование самолета	Пассажиры

Записи, для которых условие выполняется, это следующие:

ИЛ 86	АА1
-------	-----

АА1	Петров
-----	--------

Ту 154	АА2
--------	-----

АА1	Чемисов
-----	---------

Записи, неудовлетворяющие условию соединения:

Из таблицы №1

ИЛ 62	АА3
-------	-----

?	Null
---	------

Из таблицы №2

Null	?
------	---

АА4	Третьяков
-----	-----------

Теперь рассмотрим варианты соединения

1) Внутреннее соединение: в результат запроса попадут только данные записей из обеих таблиц, удовлетворяющие условию соединения т.е.

ИЛ 86	Петров
ИЛ 86	Чемисов
Ту 154	Вольский

2) Левое внешнее соединение: войдут все записи из таблицы №1, из таблицы №2 войдут записи, удовлетворяющие условию соединения. Для тех записей результата запроса, для которых не выполнялось условие соединения, в полях, где должны быть данные из таблицы №2, будут стоять значения Null.

ИЛ 86	Петров
ИЛ 86	Чемисов
Ту 154	Вольский
ИЛ 62	Null

3) Правое внешнее соединение

Правое внешнее соединение обратно левому.

ИЛ 86	Петров
ИЛ 86	Чемисов
Ту 154	Вольский
Null	Третьяков

4) Полное внешнее соединение.

В результат запроса войдут все записи: и те, для которых выполнялось условие соединения, и данных из обеих таблиц, которые не удовлетворяли условиям соединения в обеих таблицах.

ИЛ 86	Петров
ИЛ 86	Чемисов
Ту 154	Вольский
ИЛ 62	Null
Null	Третьяков

Пример:

```

Выбрать
    Номен.Наименование, База
Из Справочник.Номенклатура Номен
    Левое Внешнее Соединение
Справочник.ЕдиницыИзмерения ЕдИзм
По
    Номен.ОснЕдиницаИзмерения=ЕдИзм.Ссылка

```

VIII Упорядочивание результатов запроса

Когда необходимо упорядочить записи по определенному правилу, то в данном случае может пригодиться конструкция языка запросов «Упорядочить По»

```

Выбрать
    *
Из
    Документ.ПоступлениеТоваров
Упорядочить По Контрагент.Наименование Возр, Номер Убыв

```

Возможные варианты упорядочивания: «Возр», «Убыв», «Иерархия» (упорядочивание по иерерхии возможно только по таблицам с иерархией. Вместо имен полей, по которым производится упорядочивание, можно указывать их псевдонимы.

IX Итоги в запросе

Для получения итогов в результате запроса в тексте запроса необходимо определить конструкцию «Итоги». Итоги добавляются в результат запроса как итоговые строки.

```

Выбрать
    Номенклатура, Цена, Количество, Сумма
Из
    Документ.ПоступлениеТоваров.Товары
Итоги Сумма (Количество), Сумма (Сумма)
По Номенклатура

```

X Передача параметров в запрос

Очень часто возникает задача передачи каких-либо значений (параметров) в запрос. Например, это могут быть значения условий, накладываемых на запрос. В таком случае в тексте запроса имя параметра прописывается следующим образом:

```

Где Номенклатура=&ВыбраннаяНоменклатура
...
Запрос.УстановитьПараметр («ВыбраннаяНоменклатура», ВыбрТовар);

```

Выполнение и работа с запросами во встроенном языке

Во встроенном языке есть специальный набор объектов, чтобы осуществлять формирование запросов, выборку и обработку результатов запросов, обход записей запроса и т. д.

Приведем типичный пример использования запроса.

```
// Создадим Запрос
Запрос = Новый Запрос ("ВЫБРАТЬ Товар.Наименование
Наименование,
    |Товар.Родитель.Наименование НаименованиеРодителя
    |ИЗ Справочник.Товары Товар");
// Выполним запрос и запишем результат в переменную
// РезультатЗапроса
РезультатЗапроса = Запрос.Выполнить ();
// Получим выборку из результата запроса.
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать ();
// Пока в выборке есть записи ...
Пока ВыборкаДетальныеЗаписи.Следующий () Цикл
    // ... выведем в окно сообщений поля из результата.
    Товар = ВыборкаДетальныеЗаписи.Наименование;
    Родитель =
ВыборкаДетальныеЗаписи.НаименованиеРодителя;
    Сообщить ("Товар: " + Товар + " Родитель: " +
Родитель);
КонецЦикла;
```

Как видно из этого примера, работа с запросом ведется при помощи трех основных объектов:

- **Запрос** – объект, выполняющий сам запрос. Представлен в примере переменной с именем **Запрос**.
- **РезультатЗапроса** - объект, содержащий полученные при выполнении запроса данные. Представлен в примере переменной с именем **РезультатЗапроса**.
- **ВыборкаИзРезультатаЗапроса** - объект, позволяющий обходить (т. е. перебирать) записи из результата. Представлен в примере переменной с именем **Выборка**.

Тема 6. Методы и инструментальные средства бизнес-аналитики. Разработка сложных отчетов с использованием системы компоновки данных

Создание отчетов с помощью СКД – основные понятия и элементы. Основные принципы работы СКД. Понятия схемы компоновки и макета компоновки. Описание основных элементов схемы компоновки: наборы данных, поля, вычисляемые поля, ресурсы, параметры.

Система компоновки данных (СКД) — мощный механизм платформы «1С:Предприятие» и новее) для разработки отчетов.

Система была создана для того, чтобы декларативно создавать отчеты. Отчет создается без необходимости писать программный код. С помощью определенного конструктора и настроек, произведенных в нем, программист задает желаемый результат, система компоновки данных это понимает и выводит этот результат пользователю. Если потребовалось что-то изменить в отчете, достаточно вновь обратиться к конструктору, внести необходимые изменения, и измененный отчет готов.

В платформе «1С:Предприятие» механизм «Система компоновки данных» отвечает за построение отчетов на основе их декларативного описания.

Система компоновки данных состоит из набора элементов, каждый из которых имеет свое декларативное описание, возможность программного доступа и сериализации, то есть сохранения в XML или загрузки из XML. Это позволяет гибко управлять всеми этапами выполнения отчета.

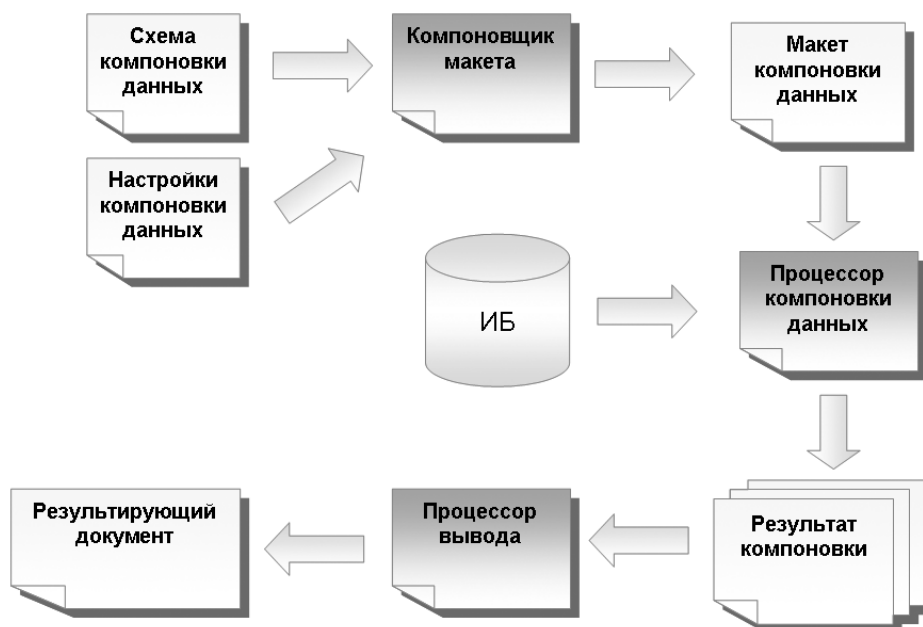


Рис. 6.1. Схема взаимодействия объектов системы компоновки данных

На рисунке 6.1. представлены этапы формирования отчета с использованием СКД. Разбивка на этапы имеет не логический, а скорее физический характер: различные этапы компоновки данных могут выполняться на различных компьютерах, в различных модулях. Разработчик в каждый из них может программно или интерактивно вмешаться.

Этапы формирования отчета:

1. Разработчик создает схему компоновки данных и стандартные настройки компоновки, которые содержат полную информацию о компоновке.

2. На основе схемы компоновки данных и настроек компоновки компоновщик макета создает макет компоновки данных, который является уже готовым заданием для процессора компоновки.

3. Процессор

компоновки данных в соответствии с макетом компоновки извлекает данные из информационной базы, агрегирует, оформляет эти данные и формирует результат компоновки. Результат компоновки состоит из массива элементов, которые можно получать последовательно, но в большинстве случаев результат компоновки получается целиком.

4. Процессор вывода обрабатывает результат компоновки и выводит его в различные форматы:

- табличный документ,
- HTML-документ,
- таблица значений,
- дерево значений.

На основе одной схемы компоновки данных может быть получено множество различных отчетов. Применение настроек компоновки данных, созданных разработчиком или измененных пользователем, к схеме компоновки данных и определяет конечный результат. Этот процесс может быть представлен с помощью следующей схемы (рис. 6.2).

Схема компоновки данных

Основой для компоновки отчета является схема компоновки данных. Создание схемы компоновки данных может быть выполнено:

- визуально при помощи конструктора схемы компоновки данных;
- визуально при помощи любого редактора, позволяющего редактировать текст XML;
- программно при помощи объектов встроенного языка системы «1С:Предприятие».

Схема компоновки данных содержит следующие разделы: наборы данных, связи наборов данных, вычисляемые поля, ресурсы, параметры, макеты, вложенные схемы, настройки.

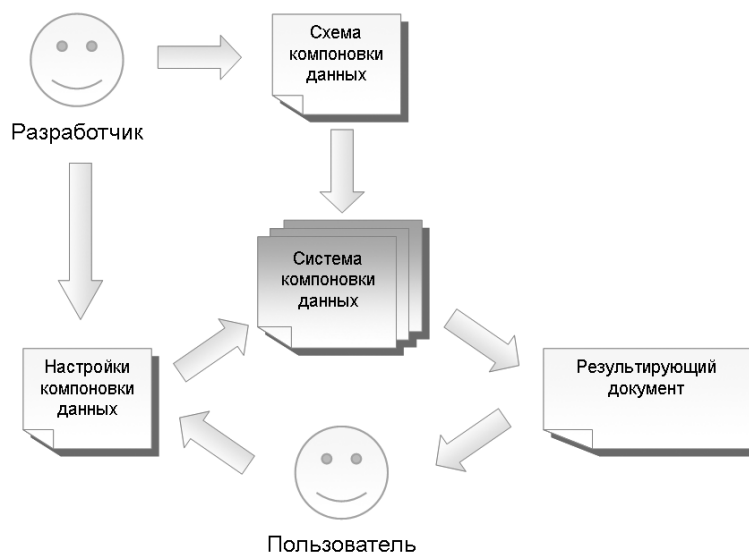


Рис. 6.2. Схема разработки и использования отчета с помощью системы компоновки данных

Вычисляемые поля

Отчет может содержать не только те поля, которые описаны в наборе данных, но и дополнительные поля, созданные разработчиком и вычисляющиеся на их основе. Значения этих полей получаются в результате вычисления выражений, написанных разработчиком в схеме компоновки данных, или в результате выполнения функций, описанных в общих модулях конфигурации. В приведенном примере выводится вычисляемое поле Прибыль как разница между полями СтоимостьПриход и СтоимостьРасход (рис. 6.3).

Продажа системных блоков

Параметры: Начало периода: 01.12.2011
Конец периода: 31.12.2011

Склад	Количество	Количество	Конечный	Стоимость	Стоимость	Прибыль
Системный блок	Приход	Расход	остаток	Приход	Расход	
Основной	2	2		17 500,00	20 000,00	2 500,00
Компьютер для дома	1	1		9 000,00	10 000,00	1 000,00
Компьютер для офиса	1	1		8 500,00	10 000,00	1 500,00
Итого	2	2		17 500,00	20 000,00	2 500,00

Рис. 6.3. Вывод в отчете вычисляемого поля

Ресурсы Система

компоновки данных может рассчитывать и добавлять в отчет записи, которые называются итоговыми. Этим записей в явном виде нет в базе данных, но они получаются в результате исполнения отчета, на основании некоторой

Остатки номенклатуры

Склад	Начальный	Приход	Расход	Конечный
Номенклатура	остаток			остаток
Основной		13,000	9,000	4,000
Материнская плата		5,000	3,000	2,000
Память		5,000	3,000	2,000
Жесткий диск		3,000	3,000	
Розничный		24,000	7,000	17,000
Монитор LCD		4,000	2,000	2,000
Веб-камера		10,000	3,000	7,000
Микрофон		10,000	2,000	8,000
Итого		37,000	16,000	21,000

Рис. 6.4. Ресурсы

обработки детальных записей, которую выполняет система компоновки данных.

Номенклатура	Основной			Розничный			Итого		
	Приход	Расход	Конечный остаток	Приход	Расход	Конечный остаток	Приход	Расход	Конечный остаток
Веб-камера				10,000	3,000	7,000	10,000	3,000	7,000
Жесткий диск	3,000	3,000					3,000	3,000	
Материнская плата	5,000	3,000	2,000				5,000	3,000	2,000
Микрофон				10,000	2,000	8,000	10,000	2,000	8,000
Монитор LCD				4,000	2,000	2,000	4,000	2,000	2,000
Память	5,000	3,000	2,000				5,000	3,000	2,000
Итого	13,000	9,000	4,000	24,000	7,000	17,000	37,000	16,000	21,000

Рис. 6.5. Вывод ресурсов в таблице

Эти групповые итоговые данные называются ресурсами схемы компоновки данных. В приведенном примере детальные записи группируются по складам и выводятся групповые и общие итоги отчета (рис. 6.4-6.5).

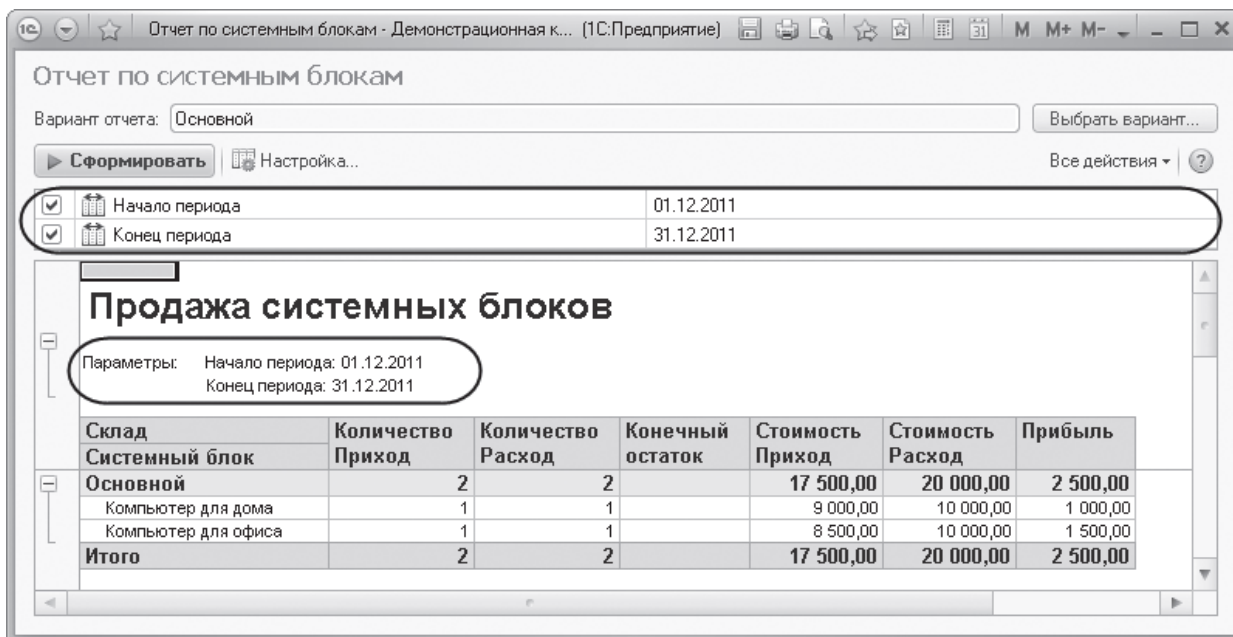


Рис. 6.6. Вывод в отчете параметров отчетного периода

Параметры

Параметры являются критериями получения данных для отчета. Параметры могут быть явно определены в запросе, например, вид номенклатуры (&ВидНоменклатуры), а могут быть параметрами виртуальных таблиц базы данных, например, начало и конец отчетного периода (рис. 6.6).

Макеты. По умолчанию внешний вид отчета формируется системой автоматически. Но разработчик может задать собственное расположение и оформление различных областей отчета. Для этого в схеме компоновки данных создаются predefined макеты. В приведенном примере группировка по номенклатуре оформлена с помощью predefined макета (рис. 6.7).

Склад	Начальный остаток	Приход	Расход	Конечный остаток
Основной		13,000	9,000	4,000
Жесткий диск		3	3	
Материнская плата		5	3	2
Память		5	3	2
Розничный		24,000	7,000	17,000
Веб-камера		10	3	7
Микрофон		10	2	8
Монитор LCD		4	2	2
Итого		37,000	16,000	21,000

Рис. 6.7. Макет группировки «Номенклатура»

Часто бывает нужно использовать данные одного отчета внутри другого. Разработав схему компоновки одного отчета, можно затем многократно использовать ее в других отчетах, связав родительский и вложенный отчеты по общему полю.

Параметры схемы компоновки

Параметры создаются автоматически при сохранении текста запроса источника данных. Можно добавить дополнительные параметры вручную, если это необходимо.

Таблица параметров имеет следующие реквизиты:

Имя – имя параметра, заданное в тексте запроса.

Заголовок – текстовое представление параметра в настройках компоновки.

Тип – тип значений параметра.

Доступные значения – список доступных значений в рамках указанного типа. Можно определить произвольные синонимы доступных значений.

Значение – значение параметра по умолчанию. Можно отключить видимость таких параметров в настройках компоновки.

Доступен список значений – флаг, позволяющий задать список значений для параметра.

Выражение – задается выражение на языке выражений СКД. В выражении можно использовать функции СКД, и имена параметров, текущего или других (указание параметров должно начинаться с символа «&»). Можно использовать для приведения заданного параметра к определенному виду, или для расчета параметра на основании других параметров.

Параметр функциональной опции – необходимо указать параметр функциональной опции, если требуется передать его системе компоновки данных. Это необходимо для определения видимости полей данных в режиме 1С-Предприятие, которые привязаны к параметризуемой ФО.

Значения не параметризуемых ФО также учитывается СКД и влияют на видимость полей в пользовательском режиме.

Включать в доступные поля – флаг, определяющий видимость параметра в настройках компоновки, на закладках: «Поля», «Отборы», «Сортировка».

Ограничение доступности – флаг, ограничивающий видимость параметра на закладке «Параметры» в настройках компоновки.

Запрещать не заполненные значения – флаг, запрещающий не заполненные значения параметра.

Использование – определяется использование параметра в итоговом запросе макета компоновки. Значение «Авто» - позволяет не включать отбор по данному параметру в итоговом запросе, если он не задан и отбор по нему указан в расширении запроса схемы компоновки (в фигурных скобках). Значение «Всегда» говорит о том, что параметр включается в итоговый запрос макета компоновки в любом случае. В этом случае, в пользовательском режиме, использование данного параметра отключить нельзя.

Параметры редактирования – настройки редактирования поля параметра, аналогичны настройкам для основных и вычисляемых полей компоновки (описаны выше).

Работа с системой компоновки может производиться и ***средствами встроенного языка*** [5]. Каждый этап компоновки данных может быть реализован программным образом. Это весьма трудоемкое занятие, и делать это целиком вряд ли понадобится, но иногда бывает необходимо заполнить наборы

данных, сформировать и вывести отчет, изменить его настройки и т. п. с помощью встроенного языка.

Система компоновки данных позволяет создать для отчета несколько схем компоновки данных. Средствами встроенного языка нужную схему компоновки можно получить по имени и программно выполнить отчет на ее основе:

```
СхемаКомпоновкиДанных =
```

```
Отчеты.СписокНоменклатуры.ПолучитьМакет ("ОсновнаяСхемаКомпоновкиДанных");
```

Средствами встроенного языка можно создать и описать саму схему компоновки данных.

```
КД = Новый СхемаКомпоновкиДанных;
```

```
ИД = СКД.ИсточникиДанных.Добавить ();
```

```
ИД.Имя = "ИсточникДанных";
```

```
ИД.ТипИсточникаДанных = "Local";
```

```
НаборДанных = СКД.НаборыДанных.Добавить (Тип ("НаборДанныхЗапросСхемыКомпоновкиДанных"));
```

```
НаборДанных.Имя = "НаборДанных1";
```

```
НаборДанных.Запрос = " ВЫБРАТЬ ЦеныСрезПоследних.Период, "+  
"ЦеныСрезПоследних.Номенклатура, ЦеныСрезПоследних.Цена "+  
"ИЗ РегистрСведений.Цены.СрезПоследних КАК ЦеныСрезПоследних";
```

```
НаборДанных.ИсточникДанных = "ИсточникДанных";
```

Затем в набор данных добавляются поля Номенклатура, Период и Цена и задается его формат без вывода дробной части. Далее в набор данных добавляется поле ресурсов Цена и задается выражение для его расчета.

```
Поле1 = НаборДанных.Поля.Добавить (Тип ("ПолеНабораДанныхСхемыКомпоновкиДанных"));
```

```
Поле1.Поле = "Номенклатура";
```

```
Поле1.ПутьКДанным = "Номенклатура";
```

```
Поле1.Заголовок = "Номенклатура";
```

```
Поле2 = НаборДанных.Поля.Добавить (Тип ("ПолеНабораДанныхСхемыКомпоновкиДанных"));
```

```
Поле2.Поле = "Период";
```

```
Поле2.ПутьКДанным = "Период";
```

```
Поле2.Заголовок = "Период";
```

```
Поле3 = НаборДанных.Поля.Добавить (Тип ("ПолеНабораДанныхСхемыКомпоновкиДанных"));
```

```
Поле3.Поле = "Цена";
```

```
Поле3.ПутьКДанным = "Цена";
```

```
Поле3.Заголовок = "Цена";
```

```
Поле3.Оформление.УстановитьЗначениеПараметра ("Формат", "ЧЦ=6; ЧДЦ=0");
```

```
ПолеРесурса = СКД.ПоляИтого.Добавить ();
```

```
ПолеРесурса.ПутьКДанным = "Цена";
```

```
ПолеРесурса.Выражение = "Сумма (Цена)";
```

В набор данных также можно добавить вычисляемые поля, параметры и т. д.

Стандартные настройки – это настройки, которые делает разработчик в схеме компоновки данных. Средствами встроенного языка можно получить и изменить стандартные настройки схемы компоновки данных.

В приведенном ниже примере создается группировка по полю Номенклатура и вложенная в нее группировка по полю Период. Затем создается автополе для списка полей и упорядочивания этих группировок. Затем в общий список выбранных полей добавляется поле ресурса Цена. В заключение задается заголовок и макет оформления отчета [5].

```

Настройки = СхемаКомпоновкиДанных.НастройкиПоУмолчанию;
ГруппировкаНоменклатура = Настройки.Структура.Добавить (Тип ("ГруппировкаКомпоновкиДанных"));
ГруппировкаНоменклатура.Использование = Истина;
ГруппировкаПериод = ГруппировкаНоменклатура.Структура.Добавить (Тип ("ГруппировкаКомпоновкиДанных"));
ГруппировкаПериод.Использование = Истина;
ПолеНоменклатура = ГруппировкаНоменклатура.ПоляГруппировки.Элементы.Добавить (Тип ("ПолеГруппировкиКомпоновкиДанных"));
ПолеНоменклатура.Использование = Истина;
ПолеНоменклатура.Поле = Новый ПолеКомпоновкиДанных ("Номенклатура");
ПолеПериод = ГруппировкаПериод.ПоляГруппировки.Элементы.Добавить (Тип ("ПолеГруппировкиКомпоновкиДанных"));
ПолеПериод.Использование = Истина;
ПолеПериод.Поле = Новый ПолеКомпоновкиДанных ("Период");
ВыбранныеПоляДляНоменклатуры = ГруппировкаНоменклатура.Выбор.Элементы.Добавить (Тип ("АвтоВыбранноеПолеКомпоновкиДанных"));
ВыбранныеПоляДляНоменклатуры.Использование = Истина;
ПорядокДляНоменклатуры = ГруппировкаНоменклатура.Порядок.Элементы.Добавить (Тип ("АвтоЭлементПорядкаКомпоновкиДанных"));
ПорядокДляНоменклатуры.Использование = Истина;
ВыбранныеПоляДляПериода = ГруппировкаПериод.Выбор.Элементы.Добавить (Тип ("АвтоВыбранноеПолеКомпоновкиДанных"));
ВыбранныеПоляДляПериода.Использование = Истина;
ПорядокДляПериода = ГруппировкаПериод.Порядок.Элементы.Добавить (Тип ("АвтоЭлементПорядкаКомпоновкиДанных"));
ПорядокДляПериода.Использование = Истина;
ВыбранныеПоля = Настройки.Выбор.Элементы.Добавить (Тип ("ВыбранноеПолеКомпоновкиДанных"));
ВыбранныеПоля.Использование = Истина;
ВыбранныеПоля.Заголовок = "Цена";
ВыбранныеПоля.Поле = Новый ПолеКомпоновкиДанных ("Цена");
Настройки.ПараметрыВывода.УстановитьЗначениеПараметра ("Заголовок", "Цены комплектующих");
Настройки.ПараметрыВывода.УстановитьЗначениеПараметра ("МакетОформления", "Зеленый");

```

Работа с параметрами схемы компоновки.

Часто бывает нужно задать значения параметров схемы компоновки данных по умолчанию перед формированием отчета. Для этого необходимо использовать обработчик события формы отчета ПередЗагрузкойВариантаНаСервере, чтобы в нем изменить параметры настроек перед их загрузкой в компоновщик [5].

&НаСервере

Процедура ПередЗагрузкойВариантаНаСервере (Настройки)

```
Настройки.ПараметрыВывода.УстановитьЗначениеПараметра ("Заголовок",
    "Продажа системных блоков");
Настройки.ПараметрыДанных.УстановитьЗначениеПараметра ("НачалоПериода",
    '20111201');
Настройки.ПараметрыДанных.УстановитьЗначениеПараметра ("КонецПериода",
    '20111231');
```

КонецПроцедуры

В параметре Настройки в обработчик передаются настройки варианта отчета, имеющие тип НастройкиКомпоновкиДанных. В данном обработчике методом УстановитьЗначениеПараметра() изменен заголовок отчета, а также установлены значения параметров начала и конца отчетного периода.

Однако если настройки являются пользовательскими (например, это параметры отчетного периода) и пользователь хоть раз задавал их значения в форме отчета, то нужно использовать обработчик события ПередЗагрузкойПользовательскихНастроекНаСервере, который вызывается после загрузки настроек варианта отчета. При этом восстанавливаются пользовательские настройки, сделанные пользователем при последнем формировании отчета и автоматически сохраненные платформой. Пользовательские настройки накладываются на настройки варианта и загружаются в компоновщик настроек

&НаСервере

Процедура ПередЗагрузкойПользовательскихНастроекНаСервере (Настройки)

```
Настройки.Элементы[0].Значение = '20111201';
Настройки.Элементы[0].Использование = Истина;
Настройки.Элементы[1].Значение = '20111231';
Настройки.Элементы[1].Использование = Истина;
```

КонецПроцедуры

В параметре Настройки в обработчик передаются настройки варианта отчета, имеющие тип ПользовательскиеНастройкиКомпоновкиДанных. Так как параметра всего два, то можно обращаться к ним по индексу (0, 1) в коллекции элементов пользовательских настроек. Мы устанавливаем значения параметров начала и конца отчетного периода и задаем для них признак использования. Эти настройки загружаются в компоновщик настроек, и отчет формируется с заданными параметрами.

Редактирование настроек схемы компоновки

Стандартные настройки схемы компоновки могут изменяться пользователем в режиме 1С:Предприятие. При создании формы отчета последние пользовательские настройки загружаются в компоновщик настроек отчета. Средствами встроенного языка, используя свойства основного реквизита отчетной формы (Отчет.КомпоновщикНастроек), можно получить и изменить эти настройки. В приведенном примере создается отбор по номенклатуре из группы Комплектующие и включается вывод информации об отборе, затем

измененные настройки загружаются в компоновщик настроек отчета, и формируется отчет.

```
Настройки = Отчет.КомпоновщикНастроек.Настройки;
```

```
ЭлементОтбора = Настройки.Отбор.Элементы.Добавить (Тип ("ЭлементОтбораКомпоновкиДанных"));
```

```
ЭлементОтбора.ЛевоеЗначение = Новый ПолеКомпоновкиДанных ("Номенклатура");
```

```
ЭлементОтбора.ВидСравнения = ВидСравненияКомпоновкиДанных.Иерархии;
```

```
ЭлементОтбора.ПравоеЗначение =
```

```
    Справочники.Номенклатура.НайтиПоНаименованию ("Комплектующие");
```

```
Настройки.ПараметрыВывода.УстановитьЗначениеПараметра ("ВыводитьОтбор",  
    ТипВыводаТекстаКомпоновкиДанных.Выводить);
```

```
Отчет.КомпоновщикНастроек.ЗагрузитьНастройки (Настройки);
```

```
СкомпоноватьРезультат ();
```

Редактирование настроек не основной схемы компоновки

Иногда требуется обеспечить редактирование настроек компоновки данных не для основной схемы компоновки данных отчета либо вообще не в отчете. Для обеспечения редактирования настроек произвольной схемы компоновки данных нужно инициализировать компоновщик настроек источником доступных настроек. После инициализации компоновщика настроек он будет оперировать доступными полями схемы, источник которой ему установлен.

Для этого нужно создать источник доступных настроек для адреса схемы компоновки данных. В качестве адреса схемы компоновки данных может выступать адрес во временном хранилище, в которое поместили схему компоновки данных, или адрес значения в записи базы данных.

```
АдресСхемы = ПоместитьВоВременноеХранилище (Схема, УникальныйИдентификатор);
```

```
КомпоновщикНастроек.Инициализировать (Новый
```

```
    ИсточникДоступныхНастроекКомпоновкиДанных (АдресСхемы));
```

В данном примере во временное хранилище помещается схема компоновки данных, находящаяся в переменной Схема.

Листинг 3.60. Инициализация компоновщика настроек источником доступных настроек

```
АдресСхемы = ПолучитьНавигационнуюСсылку (Ссылка, "СхемаКомпоновкиДанных");
```

```
КомпоновщикНастроек.Инициализировать (Новый
```

```
    ИсточникДоступныхНастроекКомпоновкиДанных (АдресСхемы));
```

В данном примере схема будет получаться из базы данных, из реквизита с именем СхемаКомпоновкиДанных объекта со ссылкой Ссылка.

Вывод отчета

После того как установлена схема компоновки данных и ее настройки, можно программно сформировать и вывести отчет.

```
Процедура СформироватьОтчет () Экспорт
```

```
    СхемаКомпоновкиДанных = УстановитьСхему ();
```

```
    Настройки = УстановитьНастройки (СхемаКомпоновкиДанных);
```

```
    // В этот объект помещается информация о расшифровке данных.
```

```

ДанныеРасшифровки = Новый ДанныеРасшифровкиКомпоновкиДанных;
// Формирование макета компоновки данных с помощью компоновщика макета.
КомпоновщикМакета = Новый КомпоновщикМакетаКомпоновкиДанных;
// В макет компоновки передаются СхемаКомпоновкиДанных, ее настройки и
// ДанныеРасшифровки для заполнения информации о расшифровке.
МакетКомпоновки = КомпоновщикМакета.Выполнить (СхемаКомпоновкиДанных, Настройки,
    ДанныеРасшифровки);
// Выполнение компоновки данных с помощью процессора компоновки.
ПроцессорКомпоновкиДанных = Новый ПроцессорКомпоновкиДанных;
ПроцессорКомпоновкиДанных.Инициализировать (МакетКомпоновки, , ДанныеРасшифровки)
;
// Очистка поля табличного документа - Результат.
Результат.Очистить ();
// Вывод результатов компоновки в табличный документ.
ПроцессорВывода = Новый ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокум
ент;
ПроцессорВывода.УстановитьДокумент (Результат);
ПроцессорВывода.Вывести (ПроцессорКомпоновкиДанных);
КонецПроцедуры

```

Система компоновки данных позволяет выводить результат компоновки данных в результирующий документ. Для этого у процессора вывода реализован метод Вывести(), в качестве параметра которому передается процессор компоновки данных. В приведенном выше примере результат выводится в элемент формы – поле табличного документа. В следующем примере результат компоновки выводится не в отчетную форму, а в новый табличный документ.

```

ДокументРезультат = Новый ТабличныйДокумент;
ПроцессорВывода = Новый ПроцессорВыводаРезультатаКомпоновкиДанныхВТабличныйДокум
ент;
ПроцессорВывода.УстановитьДокумент (ДокументРезультат);
ПроцессорВывода.Вывести (ПроцессорКомпоновкиДанных);

```

Процессор компоновки данных выполняет автоматическую фиксацию заголовка таблицы, если в отчет выводится одна таблица или одна группировка (возможно, с вложенными группировками).

Вывод в дерево значений, таблицу значений для дальнейшей программной обработки. Результат компоновки можно использовать для дальнейшей программной обработки и выводить таблицу значений или дерево значений. Для этого предназначен объект ПроцессорВыводаРезультатаКомпоновкиДанныхВКоллекциюЗначений. Метод УстановитьОбъект() является аналогом метода УстановитьДокумент(). Если метод УстановитьОбъект() не был вызван, результат будет выведен в таблицу значений.

Элементы результата компоновки могут быть получены при помощи процессора компоновки данных либо сформированы любыми другими средствами.

Тема 7. Системы на основе аналитики больших данных. Интеграция приложений на платформе «1С:Предприятие» с другими информационными системами: форматы файлов обмена, планы обмена

Интеграция с другими информационными системами, форматы файлов обмена. Планы обмена

Система «1С:Предприятие 8» является открытой системой. Она предоставляет возможность для интеграции практически с любыми внешними программами и оборудованием на основе общепризнанных открытых стандартов и протоколов передачи данных.

Организация обмена данными между несколькими информационными системами является очень важной и «популярной» задачей при организации работы в компании. Очень часто такая необходимость возникает, когда организация имеет распределенную структуру. Ее подразделения могут находиться далеко друг от друга и взаимодействовать только посредством интернета. В каждом таком территориально удаленном подразделении может находиться свой сервер или компьютер, выполняющий роль сервера. На этом сервере находится, например, база данных системы «1С:Предприятие». Для полноценной работы организации между удаленными подразделениями необходимо настроить обмен данными [1].

Также может быть, что вся организация расположена в пределах одного здания, но учет ведется в различных информационных базах на основе разных программных продуктов 1С. Чаще всего это необходимо для конфиденциальности информации и для разграничения информационных потоков с целью лучшей управляемости на каждом этапе общего бизнес-процесса организации [1].

Наиболее серьезный вариант это обмен между территориально распределенными системами на основе разных продуктов 1С, но и обмен со сторонними программными решениями (не 1С).

Механизмы обмена данными - это набор средств системы «1С:Предприятие», предназначенных для организации обмена данными между различными информационными базами, а также информационными базами и внешними программными системами. Механизмы обмена данными могут быть условно разделены на два уровня [1]:

- универсальные механизмы обмена данными,
- распределенные информационные базы.

Помимо обмена между различными конфигурациями 1С, очень часто требуется организовать обмен данными с внешними программами, например, обмен 1С с банком или логистической системой, интеграция с интернет-магазином или корпоративным порталом и т.д.

«1С:Предприятие» позволяет осуществлять интеграцию с любыми внешними программами на основе различных протоколов передачи данных. С развитием платформы возможности интеграции расширяются.

Рассмотрим несколько наиболее распространенных форматов для интеграции с различными приложениями.

Веб-сервисы в 1С (Web-сервисы)

Web-сервисы – это один из механизмов платформы, используемых для интеграции с другими информационными системами. Он является средством поддержки SOA (Service-Oriented Architecture) – сервис-ориентированной архитектуры, которая является современным стандартом интеграции приложений и информационных систем.

Значительным преимуществом сервис-ориентированной архитектуры является то, что она позволяет развивать инфраструктуру предприятия однородным образом, без разрушения уже существующих решений. Ее использование позволяет минимизировать издержки за счет интеграции разнородных и унаследованных систем в современный ландшафт предприятия. Она позволяет реализовывать слабо связанные программные компоненты с тем, чтобы максимально повысить их повторную используемость.

Пример использования: обмен данными «в режиме реального времени». При изменении данных в одной из систем, участвующих в обмене, запускается обращение к веб-сервису. Формируется пакет с измененными данными, и эти данные передаются в другую систему.

Веб-сервисы широко используются для интеграции различных приложений. В платформу 1С:Предприятие включены широкие возможности для работы с web-сервисами. 1С может выступать как поставщик Веб-сервисов и как потребитель веб-сервисов, опубликованных другим приложением.

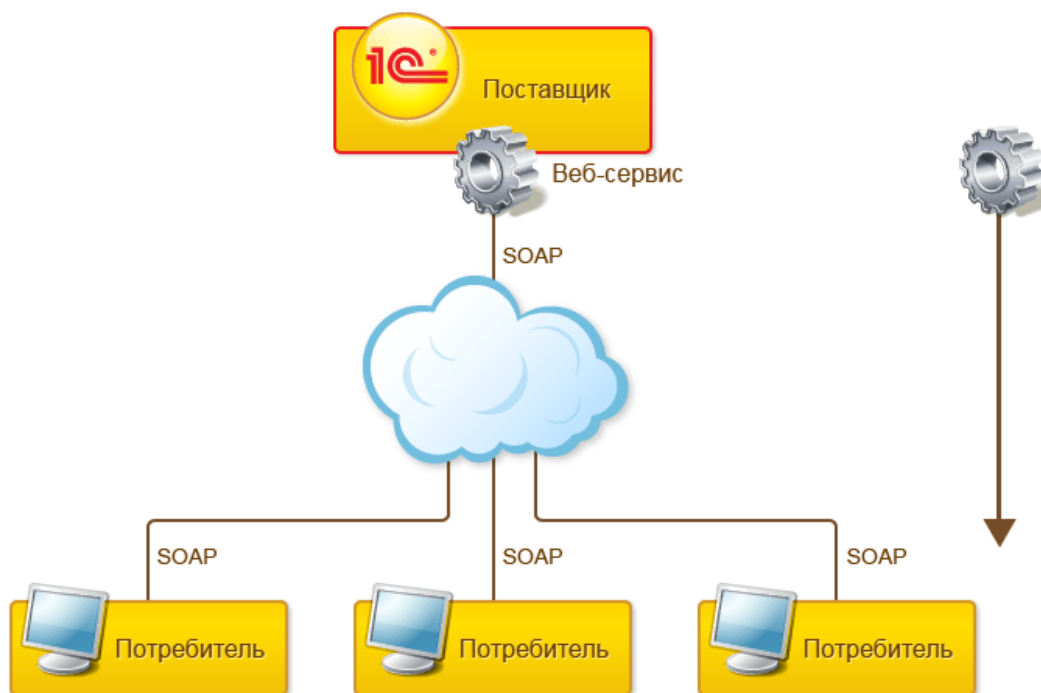


Рис. 7.1. «1С:Предприятие» — поставщик веб-сервисов

1С:Предприятие — поставщик веб-сервисов (рис. 7.1). В конфигурацию можно добавить специальный объект, — Web-сервис, — с помощью которого описать некоторую функциональность прикладного решения, например, получение списка имеющихся на некотором складе товаров, их количества и цен.

После публикации на веб-сервере такой сервис будет доступен сторонним потребителям.

В качестве потребителей могут выступать системы, использующие произвольные аппаратные и программные платформы. Технология веб-сервисов является платформенно независимой.

Программисты могут создать свой необходимый веб-сервис для решения конкретной задачи.

При использовании веб-сервиса нет потребности в предоставлении внешнему приложению доступа к информационной базе, что очень хорошо с точки зрения обеспечения безопасности данных. Внешнее приложение получает доступ к набору функций 1С, которые сами обрабатывают данные и предоставляют «наружу» конечный результат.

Если 1С передает информацию, в конфигураторе создается новый объект «веб-сервис», и программно описывается его функциональность, например, получение остатков на складах. После того как сервис будет опубликован, стороннее приложение сможет запрашивать и получать информацию о наличии требуемой номенклатуры на складах. Для публикации веб-сервиса на сервере должно быть установлено дополнительное программное обеспечение, веб-сервер. Например, это может быть бесплатный веб-сервер Apache.

1С:Предприятие — потребитель веб-сервисов (рис. 7.2).

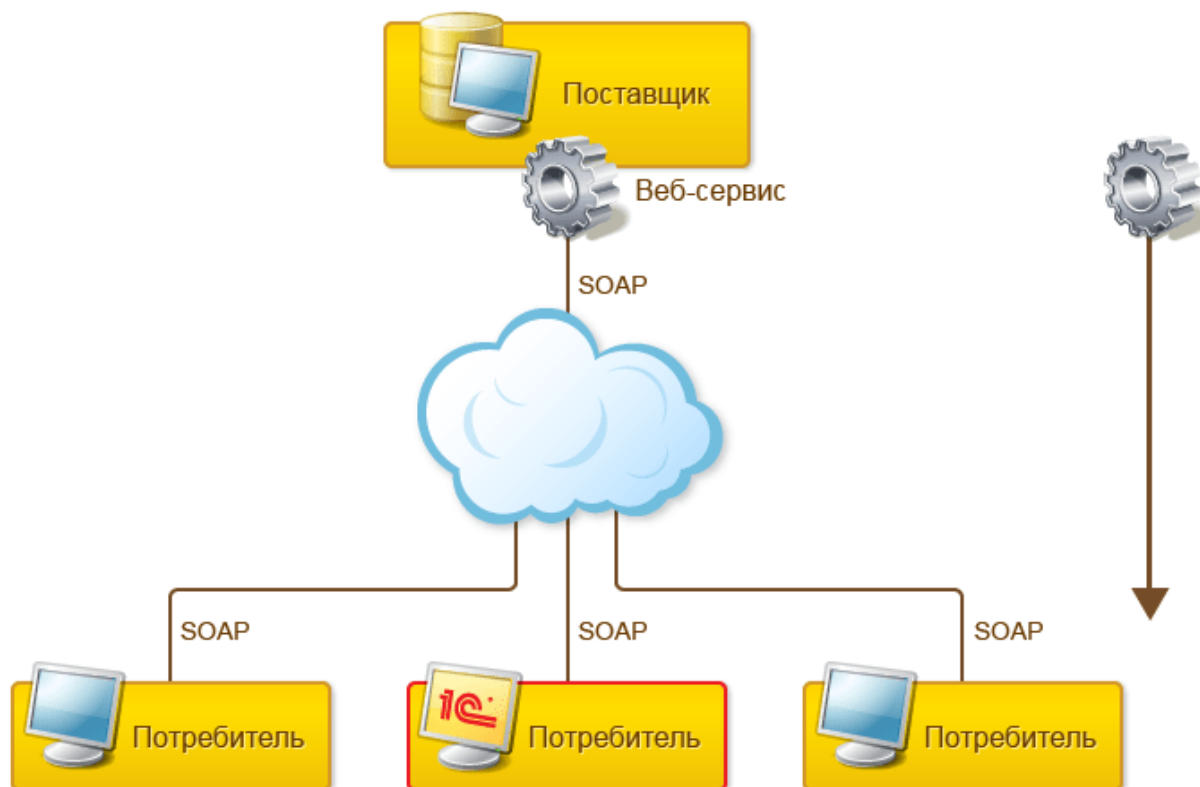


Рис. 7.2. «1С:Предприятие» — потребитель веб-сервисов

Когда 1С получает информацию, в конфигурации создается ссылка, по которой приложение 1С обращается к внешнему веб-сервису, созданному в

стороннем приложении. Получив по данной ссылке необходимые данные их обработка выполняется уже внутренними алгоритмами.

Таким образом, в прикладном решении можно описать ссылку на веб-сервис, опубликованный сторонним поставщиком. После этого прикладное решение сможет использовать данные, получаемые с помощью такого веб-сервиса, в своих внутренних прикладных алгоритмах.

WS-ссылка — это общий объект конфигурации. Она предназначена для описания в прикладном решении «статической» ссылки на некоторый внешний веб-сервис стороннего поставщика.

WS-ссылка представляет собой WSDL описание веб-сервиса (Web Services Description Language, язык описания веб-сервисов и доступа к ним, основанный на языке XML), импортированное из указанного источника. WS-ссылка недоступна для редактирования, однако можно просмотреть ее структуру и структуру типов данных, которые используются для описания параметров и возвращаемых значений.

Дальнейшая работа с такой ссылкой выполняется средствами встроенного языка. Описание ссылки на веб-сервис в дереве конфигурации, с помощью объекта WS-ссылка, удобно для редко меняющихся сервисов. Описание веб-сервиса получается один раз, при создании объекта конфигурации, и хранится в конфигурации. Поэтому все обращения к такой ссылке из встроенного языка будут выполняться быстро. Однако в случае изменения описания веб-сервиса, потребуется повторный импорт его описания в WS-ссылку.

Динамическое обращение к веб-сервисам

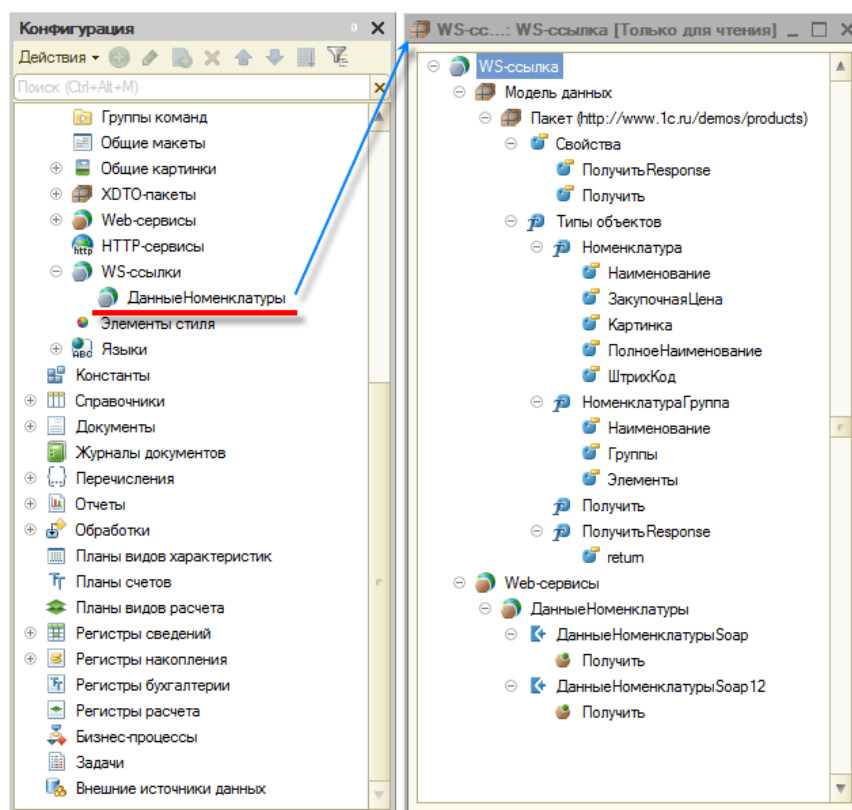


Рис. 7.3. WS-ссылка

Для работы с часто меняющимися веб-сервисами во встроенном языке предусмотрена возможность динамического считывания описания веб-сервиса и построение его прокси (Прокси – посредническое звено между компьютером, который использует абонент, и системой интернет-серверов. Если не вдаваться в терминологию, это удаленный компьютер-посредник для выхода пользователя в Интернет. Его основные задачи заключаются в трансляции всех запросов пользователя в Сеть и отправке обратно полученных ответов.). Также эта возможность позволяет вызывать веб-сервисы, расположение которых станет известно только уже в процессе выполнения программы. Однако, несмотря на все преимущества, такой способ работы медленнее, т. к. каждый раз при создании прокси веб-сервера будет тратиться время на получение описания веб-сервиса.

Ниже приведен пример создания прокси веб-сервиса и использования одного из методов веб-сервиса (рис. 7.4).

```
// Создание определения веб-сервиса по WSDL-описанию
Определение = Новый WSOпределения("http://www.host.ru/mailling?WSDL");

// Создание прокси для обращения к веб-сервису и получение фабрики XDTO
Прокси = Новый WSПрокси(Определение, "http://mail", "MailService", "Mail");
Фабрика = Прокси.ФабрикаXDTO;

// Создание объекта XDTO - параметра вызова веб-сервиса
Сообщение = Фабрика.Создать(Фабрика.Тип("http://ordering", "Order"));

// Присвоение значений свойствам объекта XDTO
Сообщение.addr = "ivanov@mail.ru";
Сообщение.subject = "В ответ на ...";
Сообщение.text = "...";

// Обращение к операции веб-сервиса
СообщениеОтправлено = Прокси.send(Сообщение);
```

Рис. 7.4. Пример создания прокси веб-сервиса и использования одного из методов веб-сервиса

XDTO – это механизм, разработанный фирмой “1С” для обмена данными с другими программными системами посредством XML, позволяющий на уровне языка 1С оперировать не узлами XML, а прикладными понятиями “Сотрудник”, “Счет” и привычными встроенными типами (“ТаблицаЗначений”, “СправочникСсылка” и т.п.).

Использование HTTP-сервисов в 1С

HTTP – протокол прикладного уровня передачи данных (изначально – в виде гипертекстовых документов).

В платформе этот протокол используется для взаимодействия клиентских приложений с веб-сервером при подключении пользователей через Интернет.

Примеры применения HTTP-сервисов такие же как и для веб-сервисов. Например, двусторонний обмен данными 1С с интернет-магазином. Со стороны

сайта в 1С передаются заказы, со стороны 1С на сайт — документы оплаты и документ, подтверждающий факт отгрузки товара.

Возможность создавать HTTP-сервисы появилась в платформе 1С:Предприятие с версии 8.3.5. Теперь 1С при помощи встроенного языка позволяет создавать запросы и сформировать ответ на запрос от внешнего приложения самостоятельно, используя возможности встроенного языка 1С.

В платформе существует возможность создания собственных произвольных HTTP-сервисов в прикладном решении (рис. 7.6).

HTTP-сервисы похожи на имеющиеся в платформе веб-сервисы, но обладают несколькими преимуществами:

- более простое создание клиентского приложения;
- уменьшенный объем передаваемых данных;
- меньшая потребность в вычислительных мощностях;
- большая нацеленность на работу в мобильных устройствах.

Первые три фактора особенно важны для приложений, работающих на мобильных устройствах.

Между веб-сервисами и HTTP-сервисами существует разница в структуре запросов, которыми обмениваются приложения: Веб-сервисы (SOAP) - POST запросы, HTTP-сервисы - GET запросы. В отличие от SOAP HTTP-сервисы поддерживают формат JSON. (POST — один из многих методов запроса, поддерживаемых HTTP протоколом, используемым во Всемирной паутине. Метод запроса POST предназначен для запроса, при котором веб-сервер принимает данные, заключённые в тело сообщения, для хранения. GET — получение ресурса. POST — создание ресурса).

Также как и при работе с веб-сервисами, для работы с HTTP-сервисами необходима установка веб-сервера (Apache или IIS).

По своему «конструктивному исполнению» HTTP-сервисы очень напоминают веб-сервисы, имеющиеся в платформе. Точно так же есть



Рис. 7.5. Использование HTTP-сервисов в 1С

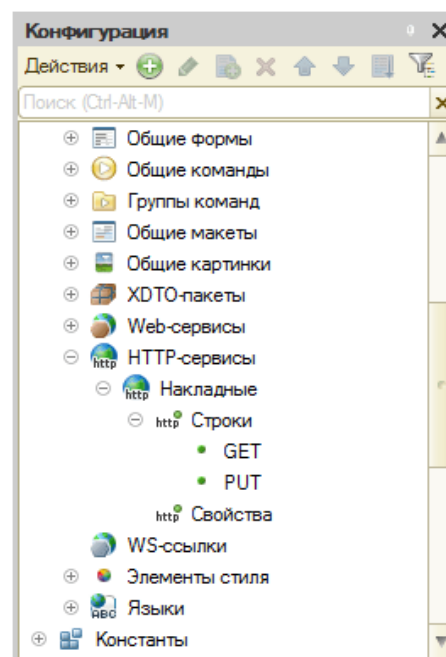


Рис. 7.6. Примеры HTTP-сервисов

специальный объект конфигурации HTTP сервис. Такие объекты добавляются в ветку Общие — HTTP-сервисы:

Каждый HTTP-сервис может содержать в себе один или несколько шаблонов (рис. 7.7). Для каждого шаблона можно создать один или несколько методов, выполняющих обработку данных:

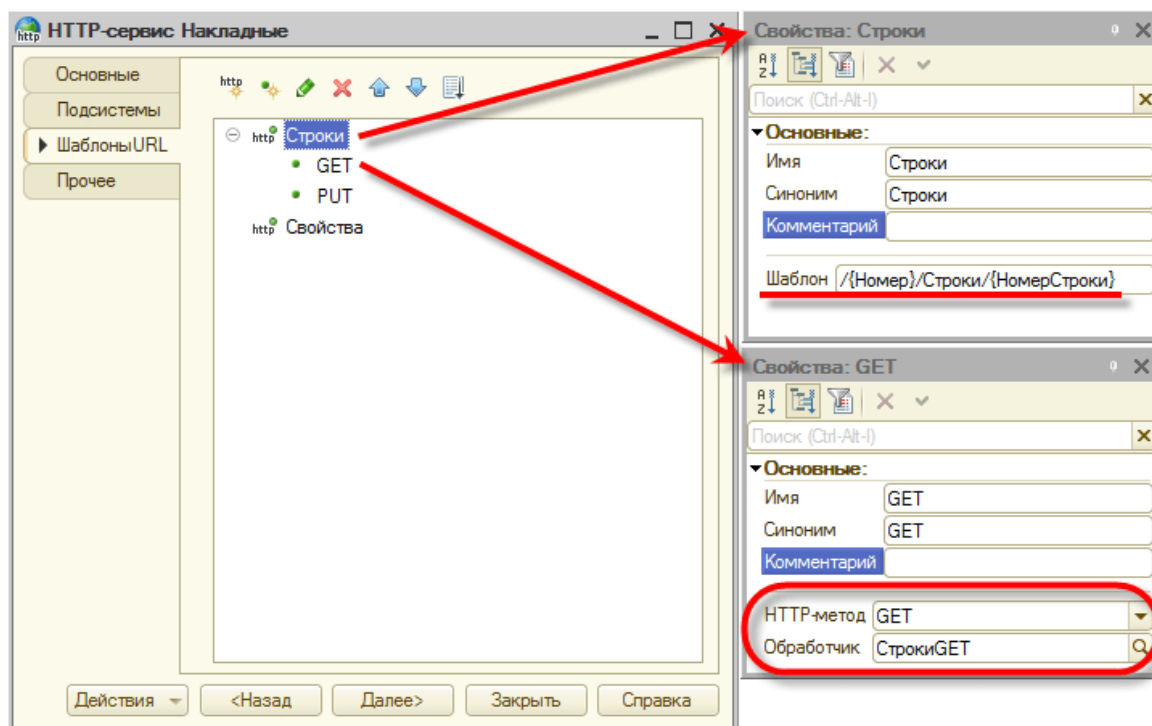


Рис. 7.7. Шаблоны HTTP-сервисов

Шаблон задаёт путь, по которому может происходить обращение к HTTP-сервису. В шаблоне можно использовать определённый набор символов, в том числе параметризованные сегменты вида {какой-то текст}.

Для каждого метода указывается, во-первых, обрабатываемый HTTP метод, а также создаётся процедура на встроенном языке, которая и будет выполнять обработку данных. Также можно указать, что будет обрабатываться не какой-то конкретный, а любой HTTP-метод из доступных.

При обращении к такому HTTP-сервису платформа сначала попытается сопоставить URL, по которому произошло обращение, с одним из имеющихся шаблонов и методов. Если сопоставить не удалось, то платформа выдаст код ответа 404 Not Found. Если подходящий метод будет найден, то платформа начнёт выполнение его обработчика, передав в него все имеющиеся в запросе данные в виде объекта встроенного языка HTTPСервисЗапрос:

Из этого объекта можно легко получить, например, параметры, содержащиеся в исходном URL, и использовать их для извлечения из базы нужных данных.

Полученные данные можно вернуть в разных форматах. Например, их можно преобразовать в XML, как на картинке выше, или даже просто в текстовую строку с разделителями.

```

HTTP-сервис Накладные: Модуль
// функция возвращает данные N-ой строки расходной накладной.
// Номер нужной накладной и номер строки содержатся в полученном HTTP запросе.
//
// Параметры:
// Запрос - HTTPСервисЗапрос
// Возвращаемое значение:
// - HTTPСервисОтвет
Функция СтрокиGET(Запрос)

    // Пример запроса:
    //     HTTP://test.server.ru/hs/Накладные.hs/000000012/Строки/1

    // Разобрать URL запроса.
    ЗапросНомерДокумента = Запрос.ПараметрыURL["Номер"];
    ЗапросНомерСтроки    = Запрос.ПараметрыURL["НомерСтроки"];

    НужныйДокумент = Документы.РасходТовара.НайтиПоНомеру(ЗапросНомерДокумента);

    // Обработка ошибочных ситуаций: документ не найден, номер не задан.
    Если НужныйДокумент = Неопределено ИЛИ НужныйДокумент.Пустая() Тогда
        Ответ = Новый HTTPСервисОтвет(404);

        Возврат Ответ;

    КонецЕсли;

    // Вернуть данные строки.
    СтрокаДокумента = НужныйДокумент.Товары[Число(ЗапросНомерСтроки) - 1];

    // Преобразовать данные строки в XML.
    ЗаписьXML = Новый ЗаписьXML;
    ЗаписьXML.УстановитьСтроку();
    ЗаписьXML.ЗаписатьОбъявлениеXML();
    ЗаписьXML.ЗаписатьНачалоЭлемента("answer");
    ЗаписьXML.ЗаписатьНачалоЭлемента("СтрокаНакладной");
    ЗаписьXML.ЗаписатьАтрибут("Товар",    СтрокаДокумента.Товар.Наименование);
    ЗаписьXML.ЗаписатьАтрибут("Цена",    Строка(СтрокаДокумента.Цена));
    ЗаписьXML.ЗаписатьАтрибут("Количество", Строка(СтрокаДокумента.Количество));
    ЗаписьXML.ЗаписатьАтрибут("Сумма",    Строка(СтрокаДокумента.Сумма));
    ЗаписьXML.ЗаписатьКонецЭлемента();
    ЗаписьXML.ЗаписатьКонецЭлемента();
    XMLСтрока = ЗаписьXML.Закрыть();

    // Передать строку в HTTP ответ.
    Ответ = Новый HTTPСервисОтвет(200);
    Ответ.УстановитьТелоИзСтроки(XMLСтрока);

    Возврат Ответ;

КонецФункции

```

Рис. 7.8. Метод HTTP-сервиса

Ответ сервиса формируется специальным объектом встроенного языка HTTPСервисОтвет, в тело которого можно поместить подготовленные данные.

Публикация HTTP-сервисов выполняется аналогично тому, как публикуются web-сервисы. Также аналогичным образом для них работает аутентификация, использование разделения данных и отладка.

Поддержка REST-интерфейса в 1С

Начиная с платформы 8.3.5 появилась возможность автоматически создавать REST-интерфейс для прикладного решения 1С (рис. 7.9). Благодаря кроссплатформенности и уникальности этого механизма, это наиболее удобное решение вопроса интеграции 1С с внешними системами. Механизм REST может использоваться и при обмене между информационными базами 1С, но для этого существуют более удобные решения.

REST-интерфейс позволяет создавать новые объекты, удалять их, читать и редактировать.

Наиболее часто REST-интерфейс применяется в следующих случаях:

- интеграции с веб-приложениями (интернет-магазины, веб-порталы и т.д.);
- обмена данными с внешним сторонним приложением;
- необходимости расширения возможности приложения 1С сторонними средствами без доработки самой конфигурации.

Типичные операции, выполняемые через REST интерфейс это:

- Получение списка документов, справочников, записей регистра сведений и т.п., возможно с фильтром;
- Получение данных элемента справочника, документа (по ссылке), данных записи независимого регистра сведений (по ключу), данных набора записей подчинённого регистра (по регистратору);
- Редактирование данных одного элемента справочника, документа и другого ссылочного объекта;
- Создание нового элемента справочника, документа, набора записей;
- Проведение одного документа, старт бизнес-процесса.

В качестве протокола доступа платформа использует протокол OData версии 3.0. Это открытый веб-протокол для запроса и обновления данных. Он позволяет оперировать данными, используя в качестве

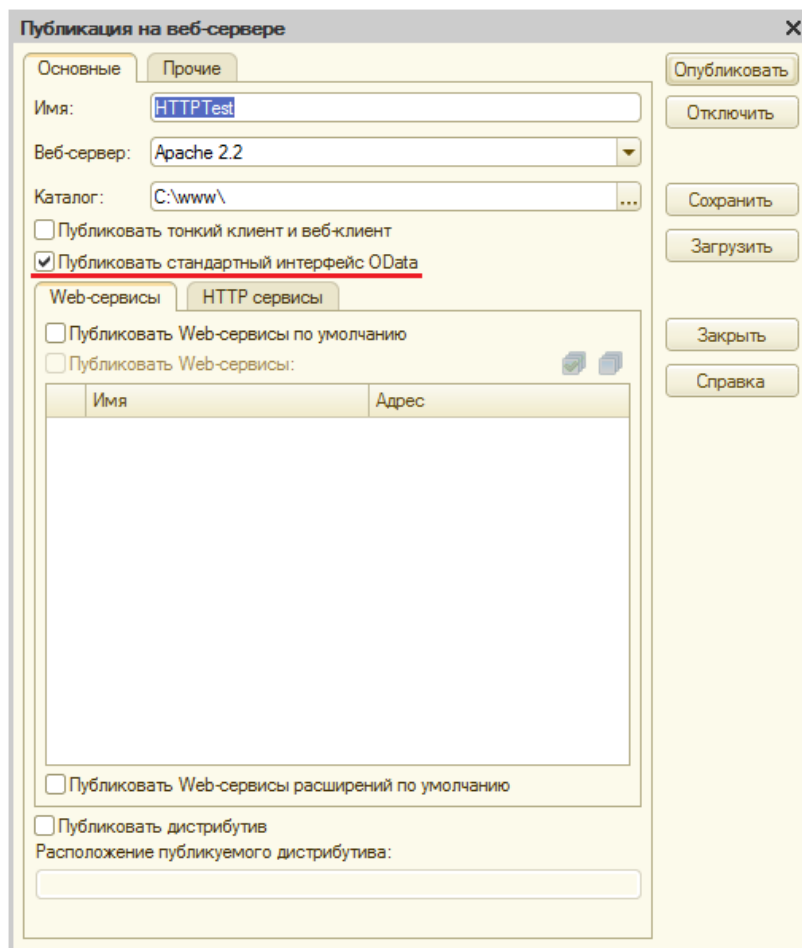


Рис. 7.9. Создание REST-интерфейса

запросов в формате Atom/XML или JSON. HTTP-команды. Получать ответы можно

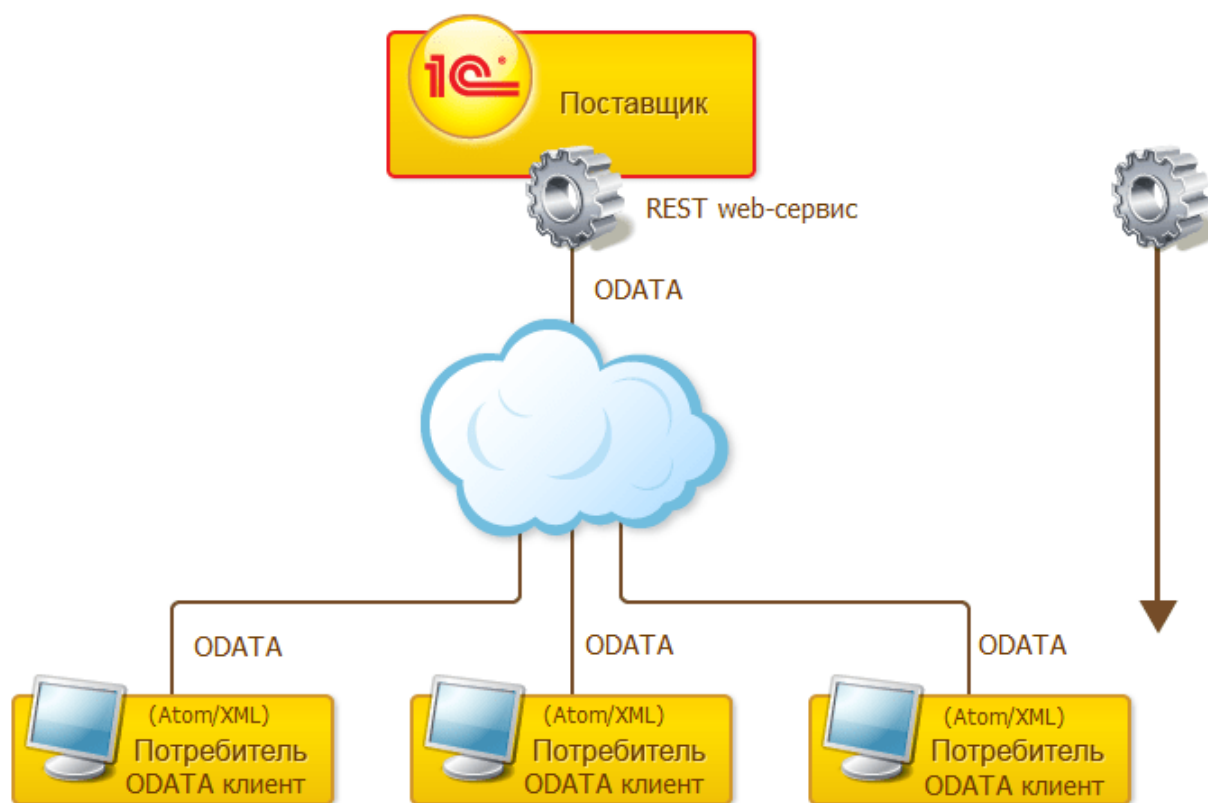


Рис. 7.10. REST web-сервис

В платформе реализована только серверная часть REST сервисов. То есть прикладное решение может автоматически поставлять свою функциональность через REST сервисы. Для взаимодействия со сторонними REST сервисами из «1С:Предприятия 8» (для организации клиентской части) можно использовать имеющиеся в платформе средства работы с HTTP.

Основная задача REST интерфейса прикладных решений заключается в интеграции со сторонними системами. И тут проблемы не возникает, ведь клиенты OData существует практически для всех значимых платформ [1]:

- Мобильные: iOS, Windows Phone, Android;
- Серверные/настольные: .NET, Java, PHP, Objective-C, Ruby, JavaScript;
- Поддержка в системах управления контентом (CMS): Drupal, Joomla.

Использовать стандартный интерфейс OData прикладного решения просто:

- В конфигураторе REST интерфейс публикуется на веб-сервере;
- После этого объекты прикладного решения становятся доступны через этот интерфейс;
- Способы аутентификации OData клиентов полностью совпадают со способами, используемыми для веб-сервисов;
- OData клиенты могут запросить через HTTP документ метаданных, описывающий доступные объекты прикладного решения;

– OData клиенты выполняют операции создания, чтения, модификации и удаления данных прикладного решения.

В REST интерфейсе доступны практически все основные объекты конфигурации: справочники, документы, константы, перечисления, планы обмена, регистры накопления, расчета, бухгалтерии и сведений, виртуальные таблицы периодического регистра сведений, регистров бухгалтерии и регистров накопления, планы счетов, видов характеристик и видов расчета, бизнес-процессы, задачи и журналы документов.

В REST интерфейсе доступны реквизиты объектов конфигурации, доступны операции создания, чтения, модификации и удаления данных, а также некоторые методы встроенного языка. Например:

- Для документа — проведение и отмена проведения;
- Для задачи — выполнение;
- Для бизнес-процесса — старт;
- Для регистра сведений — получение среза первых и среза последних;
- Для регистра накопления и регистра бухгалтерии — получение остатков, оборотов, остатков и оборотов;
- Для регистра расчета — получение данных графика, фактического периода действия, перерасчета и базы.

В случае ошибочной ситуации «1С:Предприятие 8» возвращает ответ с HTTP-статусом 4XX или 5XX. Статус 4XX говорит о неверных действиях клиента, статус 5XX — об ошибке на сервере. В случае статуса 4XX «1С:Предприятие 8» пытается помочь клиенту понять причину ошибки и может передать дополнительный внутренний код ошибки и информационное сообщение.

При чтении и записи данных с помощью REST интерфейса платформа выполняет все обычные проверки прав и вызывает обработчики событий, за исключением проверки заполнения.

При получении списков данных можно использовать стандартные условия фильтрации ODATA запросов. Например, получить товары, у которых цена меньше или равна 3,5 или больше 200:

```
GET /OData_Tests_Infobase/odata/standard.odata/Catalog_Goods?$filter=Price le 3.5 or Price gt 200
```

Применение формата XML в 1С

XML-формат является в некотором роде универсальным форматом и широко используется во всех конфигурациях 1С. 1С:Предприятие поддерживает работу с XML-документами при помощи функций встроенного языка программирования. Благодаря этому XML-формат широко используется в собственных разработках.

Формат широко используется при обмене с удаленными подразделениями и с интернет-сайтами, при загрузке выписок из банков и прайс-листов от поставщиков, при выгрузке данных в отчетные органы и т.д.

В 1С есть универсальные обработки для выгрузки данных в формате XML.

Преимущество при обмене со сторонними приложениями заключается в том что это широко распространенный формат и поддерживается большинством программных продуктов, независимо от структуры базы данных.

Также в 1С XML-формат используется при сохранении настроек отчетов и печатных форм.

Работа с XML-документами доступна непосредственно из встроенного языка системы 1С:Предприятие 8.

Имеется возможность:

- последовательно читать и записывать xml-документы:
 - преобразовывать из строки, полученной из текста элемента или значения атрибута XML, в значение в соответствии с указанным типом;
 - получать строковое представление значения для помещения в текст элемента или значение атрибута XML;
 - получить тип данных XML, соответствующий переданному в качестве параметра типу;
 - производить проверку возможности чтения из XML значения указанного типа;
 - производить проверку соответствия схеме XML при чтении XML
 - производить запись значения в формате XML;
 - возвращать тип, соответствующий типу данных XML.
- использовать модель объектного доступа к данным xml-документов (ДокументDOM), соответствующую следующим стандартам:
 - DOM Level 2;
 - XPath (DOM Level 3);
 - DOM Load and Save (DOM Level 3).
- использовать объектную модель схемы XML (СхемаXML);
- использовать канонический XML стандарт 1.1 (Canonical XML 1.1).

Используя внешнее соединение и механизмы работы с XML можно организовывать интеграцию с прикладными системами по принятым в этих

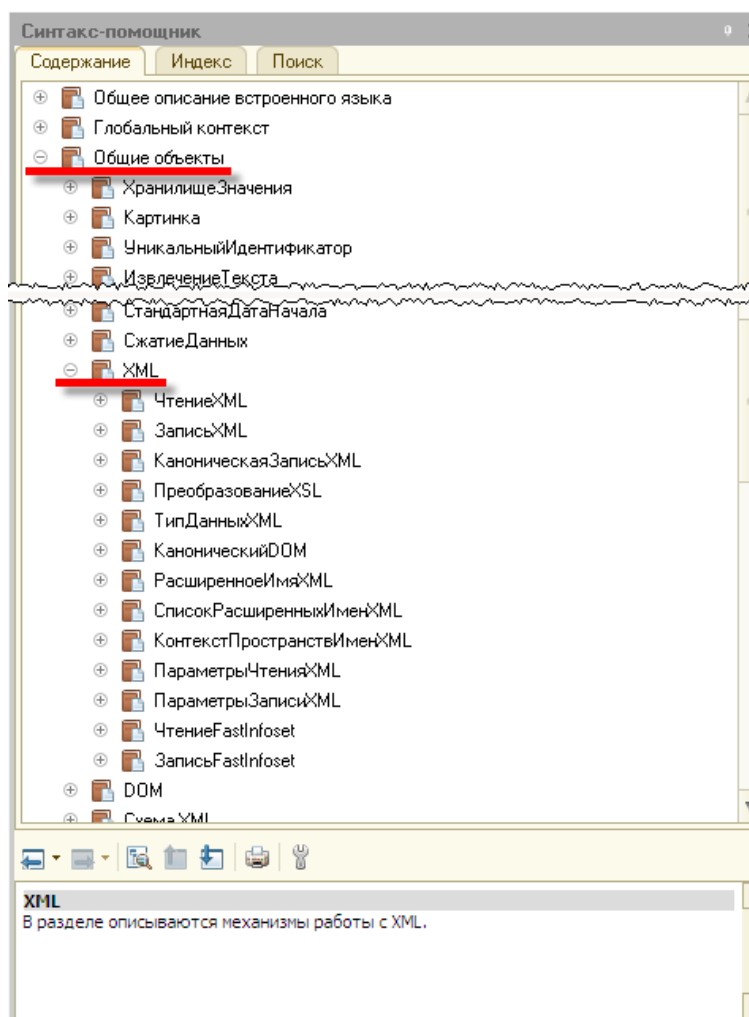


Рис. 7.11. XML

системах форматам. Для этого применяются механизмы XSL-преобразования. Например, для такой интеграции можно использовать BizTalk сервер компании Microsoft.

Поддержка JSON в 1С

JSON (JavaScript Object Notation) это текстовый формат обмена данными, широко используемый в веб-приложениях [1].

Начиная с версии 8.3.6.1977 в платформе 1С реализована поддержка формата JSON. В более ранних версиях платформы 1С с этим форматом тоже можно было работать, но теперь в 1С появились удобные стандартные средства для работы с JSON. Этот формат широко применяется в веб-приложениях и поддерживается всеми браузерами. По сравнению с XML, текстовый формат обмена JSON является более лаконичным. Кроме этого все браузеры имеют встроенные средства для работы с JSON.

Основное применение данного формате в 1С это интеграция с внешними приложениями, особенно с веб-приложениями. Формат JSON можно использовать при обмене файлами между разными приложениями 1С:Предприятие.

JSON активно используется в HTTP интерфейсах, для работы с которыми в 1С:Предприятии есть REST интерфейс приложения, автоматически генерируемый платформой, и HTTP-сервисы, которые можно создавать самостоятельно.

Все описанные выше форматы обмена данными в 1С предназначены для того чтобы подготовить запрашиваемые данные и передать их стороннему приложению. Или запросить необходимые данные у внешнего приложения, получить их и передать для обработки на основании внутренних алгоритмов приложения 1С.

Существует несколько основных сценариев использования JSON [1].

- Интеграция с внешними системами через их HTTP интерфейсы: Google Calendar, Salesforce.com, REST интерфейс 1С:Предприятия, SharePoint и т. д.;
- Организация собственного HTTP интерфейса прикладного решения;
- Обмен файлами JSON с внешними системами. Формирование конфигурационных, настроечных файлов. Использование их в процедурах обмена данными, например, с интернет-магазинами;



Рис. 7.11. Использование для интеграция BizTalk сервера компании Microsoft

– Использование файлов JSON для обмена данными между разными приложениями 1С:Предприятия.

```
// Пример записи JSON.
ЗаписьJSON = Новый ЗаписьJSON;

// Для увеличения скорости работы можно отключить автоматическую проверку
// правильности структуры записываемого документа JSON.
ЗаписьJSON.ПроверятьСтруктуру = Ложь;

// Для красоты результирующего текста можно увеличить "лесенку".
// Стандартный отступ – 1 пробел.
ПараметрыЗаписиJSON = Новый ПараметрыЗаписиJSON( , Символы.Таб);

// Либо открыть файл, в который будет выполнена запись,
// либо указать, что запись будет выполнена в строку, которую вернет метод Закрыть().
ЗаписьJSON.ОткрытьФайл("ИмяФайла", , , ПараметрыЗаписиJSON);
//ЗаписьJSON.УстановитьСтроку(ПараметрыЗаписиJSON);

// Записать массив.
ЗаписьJSON.ЗаписатьНачалоОбъекта();
ЗаписьJSON.ЗаписатьИмяСвойства("СвойствоТипаМассив");
ЗаписьJSON.ЗаписатьНачалоМассива();

// Первый элемент массива – Строка.
ЗаписьJSON.ЗаписатьЗначение("Значение строка");

// Второй элемент массива – Число.
ЗаписьJSON.ЗаписатьЗначение(12.345, Истина);

// Третий элемент массива – Булево.
ЗаписьJSON.ЗаписатьЗначение(Истина);

// Четвертый элемент массива – объект с двумя свойствами.
ЗаписьJSON.ЗаписатьНачалоОбъекта();

// Первое свойство объекта – Строка
ЗаписьJSON.ЗаписатьИмяСвойства("СвойствоТипаСтрока");
ЗаписьJSON.ЗаписатьЗначение("Значение строка");

// Второе свойство объекта – Неопределено
ЗаписьJSON.ЗаписатьИмяСвойства("СвойствоТипаНеопределено");
ЗаписьJSON.ЗаписатьЗначение(Неопределено);

ЗаписьJSON.ЗаписатьКонецОбъекта();

ЗаписьJSON.ЗаписатьКонецМассива();

ЗаписьJSON.ЗаписатьКонецОбъекта();

ЗаписьJSON.Закрыть();
//СтрокаJSON = ЗаписьJSON.Закрыть();
```

Рис. 7.12. Пример листинга

В платформе реализовано несколько слоёв работы с JSON. Самые простые и гибкие — это низкоуровневые средства потоковой записи и чтения. Более высокоуровневые и не такие универсальные — средства сериализации в JSON примитивных типов и коллекций 1С:Предприятия. И, наконец, третий слой это средства, позволяющие сериализовать/десериализовать прикладные типы 1С:Предприятия: ссылки, объекты, наборы записей и вообще любые типы, для которых поддерживается XDTO сериализация [1].

1) Средства первого слоя. Потоковое чтение и запись JSON

Объекты потоковой работы последовательно читают данные в формате JSON из файла или строки, или последовательно записывают их в файл или строку.

```

1  {
2      "СвойствоТипаМассив": [
3          "Значение строка",
4          1.2345E1,
5          true,
6          {
7              "СвойствоТипаСтрока": "Значение строка",
8              "СвойствоТипаНеопределено": null
9          }
10     ]
11 }
```

Рис. 7.13. Результат записи JSON

Таким образом, чтение и запись данных происходят без формирования всего документа в памяти.

Потоковая запись JSON может выглядеть следующим образом. Записывается массив из четырёх элементов. Три из них примитивного типа, а четвёртый элемент — это объект с двумя свойствами (рис. 7.12).

Результат такой записи представлен на рис. 7.13.

2) Второй слой. Сериализация примитивных типов и коллекций.

Вторая группа средств работы с JSON хороша тем, что избавляет от рутинной работы по чтению/записи каждого отдельного значения или свойства. При чтении документы JSON отображаются в фиксированный набор типов платформы: Строка, Число, Булево, Неопределено, Массив, ФиксированныйМассив, Структура, ФиксированнаяСтруктура, Соответствие, Дата. Соответственно, в обратную сторону, композиция объектов этих типов позволяет сформировать в памяти и быстро записать в файл структуру JSON. Таким образом, чтение и запись небольшого объема JSON заранее известной структуры можно производить немногими строчками кода.

Основное назначение этих средств в обмене информацией с внешними системами, чтении конфигурационных файлов в формате JSON.

Пример сериализации (записи) в JSON может выглядеть так (рис. 7.13).

```

// Пример записи JSON.
Структура = Новый Структура;
Структура.Вставить ("Фамилия" , "Иванов");
Структура.Вставить ("Имя" , "Иван");
Структура.Вставить ("Отчество" , "Иванович");
Структура.Вставить ("Возраст" , 40);
Структура.Вставить ("Женат" , Истина);

Телефоны = Новый Массив;
Телефоны.Добавить ("8-999-999-99-90");
Телефоны.Добавить ("8-999-999-99-91");

Структура.Вставить ("Телефоны" , Телефоны);

ЗаписьJSON = Новый ЗаписьJSON;
ЗаписьJSON.ОткрытьФайл ("Имяфайла" , , , Новый ПараметрыЗаписиJSON ( , Символы.Таб));
ЗаписатьJSON (ЗаписьJSON, Структура);
ЗаписьJSON.Закреть ();
```

Рис. 7.13. Пример сериализации (записи) в JSON

Результат записи представлен на рис. 7.14.

3) Третий слой. Сериализация прикладных типов 1С:Предприятия.

Прежде всего, и в основном, XDTO сериализацию в JSON рекомендуется использовать при обмене данными между двумя прикладными решениями 1С:Предприятия. Также этот механизм можно использовать и для обмена с внешними системами, готовыми принимать типы данных 1С:Предприятия.

Простейший код, выполняющий сериализацию элемента справочника, может выглядеть так (рис. 7.15):

```
// Подготовка к сериализации.
ЗаписьJSON = Новый ЗаписьJSON;
ПараметрыЗаписиJSON = Новый ПараметрыЗаписиJSON( , Символы.Таб );
ЗаписьJSON.ОткрытьФайл("ИмяФайла", , , ПараметрыЗаписиJSON);

// Получить объект справочника.
Значение = Справочники.Контрагенты.НайтиПоКоду("000000012").ПолучитьОбъект();

// Сериализовать объект в JSON.
СериализаторXDTO.ЗаписатьJSON(ЗаписьJSON, Значение, НазначениеТипаXML.Явное);

// Завершающие действия.
ЗаписьJSON.Закрыть();
```

Рис. 7.15. Код, выполняющий сериализацию элемента справочника

В результате будет получен файл JSON следующего содержания (рис. 7.16):

```
1 {
2   "#type": "jcfig:CatalogObject.Контрагенты",
3   "#value": {
4     "IsFolder": false,
5     "Ref": "3d710076-8d04-11dc-8ca0-000d8843cd1b",
6     "DeletionMark": false,
7     "Parent": "9d5c422c-8c4c-11db-a9b0-00055d49b45e",
8     "Code": "000000012",
9     "Description": "Мосхлеб ОАО",
10    "Регион": "9d5c422f-8c4c-11db-a9b0-00055d49b45e",
11    "Индекс": "456789",
12    "Страна": "Россия",
13    "Город": "Москва",
14    "Улица": "Петровка",
15    "Дом": "12",
16    "Телефон": "+7(999)234-78-64",
17    "ЭлектроннаяПочта": "mh@hleeb.ru",
18    "Факс": "",
19    "ВебСайт": "",
20    "ВидЦен": "9d5c4225-8c4c-11db-a9b0-00055d49b45e",
21    "ДополнительнаяИнформация": "Поставка хлеба",
22    "КонтактноеЛицо": "Громьшева П.Р.",
23    "Широта": 55.762744,
24    "Долгота": 37.618102
25  }
26 }
```

Рис. 7.16. Результат сериализации элемента справочника

```
1 {
2   "Фамилия": "Иванов",
3   "Имя": "Иван",
4   "Отчество": "Иванович",
5   "Возраст": 40,
6   "Женат": true,
7   "Телефоны": [
8     "8-999-999-99-90",
9     "8-999-999-99-91"
10  ]
11 }
```

Рис. 7.14. Результат сериализации (записи) в JSON

Использование JSON в HTTP интерфейсах приложений

А) Автоматически генерируемый REST интерфейс прикладных решений

При обращении к REST интерфейсу прикладного решения можно получать ответ в формате JSON. Есть возможность управлять объёмом передаваемой информации за счёт изменения детальности представления метаданных в выгрузке. Существуют три уровня: минимальный, средний и максимальный. Однако при этом нужно учитывать, что сокращение объёма передаваемой информации приводит к более интенсивным вычислениям на клиенте. И наоборот, когда вся информация включается в выгрузку, объём вычислений на клиенте будет минимальным.

Б) HTTP-сервисы прикладного решения

HTTP-сервисы, реализованные в прикладном решении, также могут возвращать ответ в формате JSON. При этом, при формировании тела ответа, есть возможность указать, следует ли использовать BOM (Byte Order Mark, метка порядка байтов).

В) Взаимодействие со сторонними HTTP сервисами

При взаимодействии со сторонними HTTP интерфейсами также существует возможность сформировать тело запроса в JSON, а затем установить его в HTTP запрос.

В структуре дерева метаданных «1С:Предприятие» достаточно много объектов, предназначенных для решения задач интеграции (мы рассмотрели часть из них):

- *Планы обмена* – хранит список узлов, с которыми осуществляется обмен;
- *Web-сервисы* – предоставляют возможность обращения к текущей базе 1С с помощью Web-сервисов;
- *HTTP-сервисы* – предоставляют возможность обращения к текущей базе 1С с помощью HTTP-сервисов;
- *WS-ссылки* – предназначены для описания статические ссылки на Web-сервисы сторонних поставщиков;
- *XDTO-пакеты* – объект для описания структуры данных; широко используется при работе с Web-сервисами;
- *Внешние источники данных* – предоставляет доступ к данным внешних баз, построенным не на 1С.

Можно с уверенностью сказать, что разработчики платформы уделили пристальное внимание функционалу обмена информацией с другими системами и предоставили программисту 1С широкий выбор способов интеграции прикладных решений на «1С:Предприятие» с другими информационными системами.

Планы обмена

При организации постоянного обмена может возникнуть ряд задач [1]:

- с кем будет производиться обмен (определение состава участников обмена);

- какими данными будет производиться обмен (с одной стороны, это определение перечня типов объектов; с другой стороны, определение «экземпляров»);

- определение регламента обмена (например, нумерация сообщений, адресация, процесс разрешения коллизий и т. п.). Все эти задачи в той или иной мере могут решаться с использованием функциональности планов обмена. Рассмотрим более подробно.

Планы обмена являются центром, вокруг которого группируются прочие механизмы, связанные с обменом данными. В одной конфигурации может быть определено произвольное количество планов обмена. Каждый из планов обмена определяет набор данных, которыми предполагается обмениваться в рамках данного плана обмена. Вместе с набором данных могут определяться и специфические форматы представления этих данных [2].

Элементами данных плана обмена являются узлы плана обмена, подобно тому, как элементами данных справочника являются элементы справочника. Каждый из узлов плана обмена обозначает участника обмена данными по данному плану обмена. Один из узлов (он является предопределенным) соответствует данной информационной базе, а остальные – другим участникам, с которыми данная информационная база может обмениваться данными.

Реквизиты и табличные части узла обмена могут использоваться для указания специфических данных по участнику обмена. С их помощью определяется порядок взаимодействия с данным участником, привязка его к другим объектам базы. Например, может указываться, что данный узел с точки зрения базы данных является «таким-то» складом.

В обмене данными могут участвовать [2]:

- объекты базы данных: элементы справочников, документы и т.д.
- необъектные данные: наборы записей регистров, последовательностей, контсаны;
- специальный объект встроенного языка – *Удаление объекта*.

В планах обмена можно выделить две значимые составляющие:

- инфраструктура сообщений,
- служба регистрации изменений.

Данные переносятся между узлами с помощью сообщений. Средства работы с сообщениями образуют инфраструктуру сообщений. Каждое сообщение относится к определенному плану обмена, имеет определенный узел-отправитель и определенный узел-получатель. Сообщение не может быть отправлено неизвестному узлу и не может быть принято от неизвестного узла. Каждое сообщение имеет свой собственный целочисленный номер. Сообщение оформляется как документ XML, имеющий определенную структуру [2].

С точки зрения инфраструктуры сообщений у узла существуют два Свойства [1]:

- номер отправленного сообщения,
- номер принятого сообщения.

Тема 8. Системы на основе аналитики больших данных. Интеграция приложений на платформе «1С:Предприятие» с другими информационными системами: универсальный механизм обмена данными, обмен данными в распределенных системах

Универсальный механизм обмена данными. Обмен данными в распределенных системах.

При решении различных задач бизнес-аналитики может возникнуть необходимость в создании распределенной информационной системы (в силу территориальной распределенности организации, использования разнородного программного обеспечения для решения отдельных подзадач и т. п.). Технологическая платформа «1С:Предприятие» позволяет решать такие задачи, причем распределенные информационные системы могут строиться как на основе информационных баз «1С:Предприятия», так и с задействованием совершенно других информационных систем. В качестве таких систем могут выступать как другие информационные базы «1С:Предприятия» (имеющие аналогичную или отличающуюся конфигурацию), так и программные системы, не основанные на «1С:Предприятии» [1].

В состав средств платформы, используемых для построения схем обмена данными, входят:

- объекты конфигурации План обмена;
- базовые средства работы с XML (чтения и записи документов);
- средства XML-сериализации.

При помощи *планов обмена* мы получаем информацию о том, какие элементы данных были изменены и в какой узел обмена их необходимо передать. Это возможно благодаря тому, что планы обмена содержат механизм регистрации изменений. Информация об измененных данных переносится с помощью сообщений, инфраструктура которых также поддерживается планами обмена.

XML-сериализация позволяет преобразовать объект «1С:Предприятия» в последовательность данных, представленных в формате XML. Кроме этого, XML-сериализация выполняет и обратное преобразование – преобразует последовательность данных формата XML в объект «1С:Предприятия», при условии что имеется соответствующий тип «1С:Предприятия» [1].

Дело в том, что объект обмена, являющийся в системе «1С:Предприятие» единым целым, на самом деле представляет собой совокупность данных различных типов, определенным образом связанных между собой. Например, элемент справочника кроме кода и наименования может содержать некоторое количество реквизитов различного типа и некоторое количество табличных частей, содержащих, в свою очередь, некоторое количество реквизитов

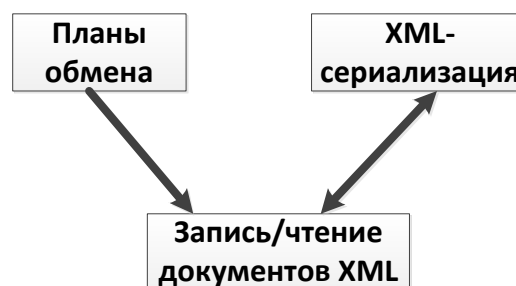


Рис. 8.1. Средства платформы, используемые для построения схем обмена данным

различного типа. В результате XML-сериализации вся эта совокупность данных представляется в виде последовательности соответствующих данных формата XML. Вследствие обратного преобразования производится «сборка» объекта при условии, что существует подходящий тип данных «1С:Предприятия» [1].

Запись и чтение документов XML обеспечивают запись/чтение документов формата XML из встроенного языка. В отличие от XML-сериализации, механизмы записи/чтения документов XML позволяют работать с данными формата XML на базовом уровне, без привязки к объектам «1С:Предприятия». В частности, они позволяют открывать файлы XML для чтения, читать данные из файлов, создавать новые файлы XML и записывать в них данные.

При помощи этих средств могут быть реализованы два механизма обмена данными [1]:

- универсальный механизм обмена данными,
- механизм распределенных информационных баз.

Универсальный механизм обмена данными позволяет создавать произвольные распределенные системы и практически не накладывает никаких ограничений на структуру создаваемой системы. Можно как связывать в единое целое базы «1С:Предприятия» с отличными друг от друга конфигурациями, так и осуществлять обмен с принципиально отличными информационными системами (базами данных).

Механизм распределенных информационных баз, напротив, предназначен для обмена данными только с идентичными конфигурациями «1С:Предприятия» и жестко регламентирует структуру создаваемой системы [1].

Таблица 8.1

Сравнение механизмов обмена данными

	Универсальный механизм обмена данными	Механизм распределенных информационных баз
В узлах плана обмена находятся	Произвольные базы данных, в том числе информационные базы «1С:Предприятия»	Только информационные базы «1С:Предприятия»
Конфигурации информационных баз	Могут быть разные (применительно к информационным базам «1С:Предприятия»)	Только идентичные
В сообщениях обмена передаются	Только изменения данных	Изменения данных, изменения конфигурации
Направление передачи	Произвольное, между двумя связанными узлами	Изменения данных – произвольное, изменения конфигурации – от главного к подчиненному узлу
Формат файлов обмена	Произвольный, чаще всего XML	XML
Создание распределенной базы и выполнение обмена	Требуется написание кода (определение порядка разрешения коллизий, стратегии распространения данных, создания начальных выгрузок, решение задачи синхронизации данных)	Может быть выполнено исключительно интерактивными средствами, без кодирования (действуют соглашения по умолчанию)
Структура распределенной системы	Произвольная. Может отсутствовать понятие главный – подчиненный (отсутствовать иерархия)	Древовидная. Любой узел (кроме корневого) имеет один главный и произвольное количество подчиненных узлов

Служба регистрации изменений. Служба регистрации изменений предназначена для регистрации изменений данных, производимых «1С:Предприятием», чтобы при обмене данными иметь возможность передавать не все данные, а только новые, измененные и удаленные [1].

При каждом изменении данных должно быть зарегистрировано, что имеются изменения, которые предстоит передать во все узлы, с которыми поддерживается обмен этими данными. При получении подтверждения приема сообщения, в котором были отправлены изменения, записи регистрации изменений должны быть удалены.

Настройка состава объектов, для которых включается регистрация изменений, производится в режиме Конфигуратор, на закладке Основные объекта конфигурации ПланОбмена (необходимо нажать кнопку Состав). Для каждого объекта можно определить свойство Авторегистрация [1].

Можно сказать, что объекты, для которых включается регистрация изменений, являются входными данными для службы регистрации изменений. Задача этой службы состоит в том, чтобы, опираясь на данный перечень объектов, отслеживать изменения объектов, их удаление и производить соответствующие записи в таблицах регистрации изменений объектов.

При этом отслеживаются ситуации повторного изменения (но об этом более подробно в разделе, посвященном именно службе регистрации изменений).

1) Доработка объектов конфигурации, участвующих в обмене.

Необходимо внести изменения в модули всех объектов, участвующих в обмене. При формировании номеров документов или кодов справочников необходимо использовать префикс. Для чего следует использовать Константу с именем *ПрефиксНумерации* (Строка,2).

Функцию формирования префикса номера вынесем в общий модуль *Обмен* (рис. 8.2). Она возвращает значение константы *ПрефиксНумерации* [1].

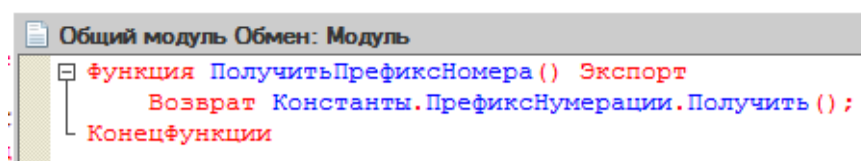


Рис. 8.2. Функция формирования префикса номера

Далее необходимо доработать каждый справочник, документ, принимающие участие в обмене. Необходимо в модуле объекта у этих объектов прописать процедуру-обработчик события ПриУстановкеНовогоКода (рис. 8.3). Прописать ее надо именно в модуле объекта, т.к. это событие возникает не для формы, а для объекта в целом. Вторым параметром вызова обработчика передается префикс, который будет заполнен в данной процедуре и использован системой для генерации кода [1].

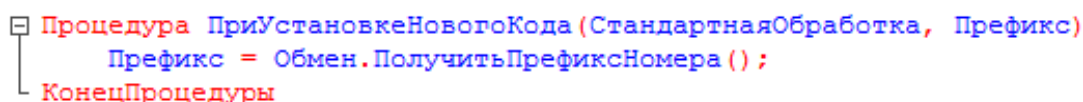


Рис. 8.3. Процедура-обработчик события ПриУстановкеНовогоКода

В обработчике события вызываем функцию общего модуля. Поскольку модуль неглобальный, то обращаемся к ней по имени модуля и имени функции (Обмен.ПолучитьПрефиксНомера();). В этой процедуре мы устанавливаем префикс равным значению константы ПрефиксНумерации.

Такие обработчики надо добавить во все справочники, документы и т.п., участвующие в обмене. После этого у всех объектов нужно увеличить длину кода с 9 до 11 символов (с учетом префикса) [1].

2) Добавление плана обмена

а) В ветви Общие дерева объектов конфигурации добавим новый объект План обмена с именем Филиалы. На закладке Данные создадим реквизит плана обмена Главный типа Булево. Он будет использоваться для разрешения коллизий при обмене данными (когда один и тот же объект обмена данными был изменен одновременно в двух узлах. В этом случае будем анализировать значение реквизита Главный и принимать изменения только в том узле, который главный, а изменения не в главном узле, если возникает коллизия, отвергаются [2].

б) Определим состав объектов, участвующих в обмене. Для этого на закладке Основные нажмем кнопку Состав (рис. 8.4). Отметим объекты для обмена [1] (рис. 8.5).

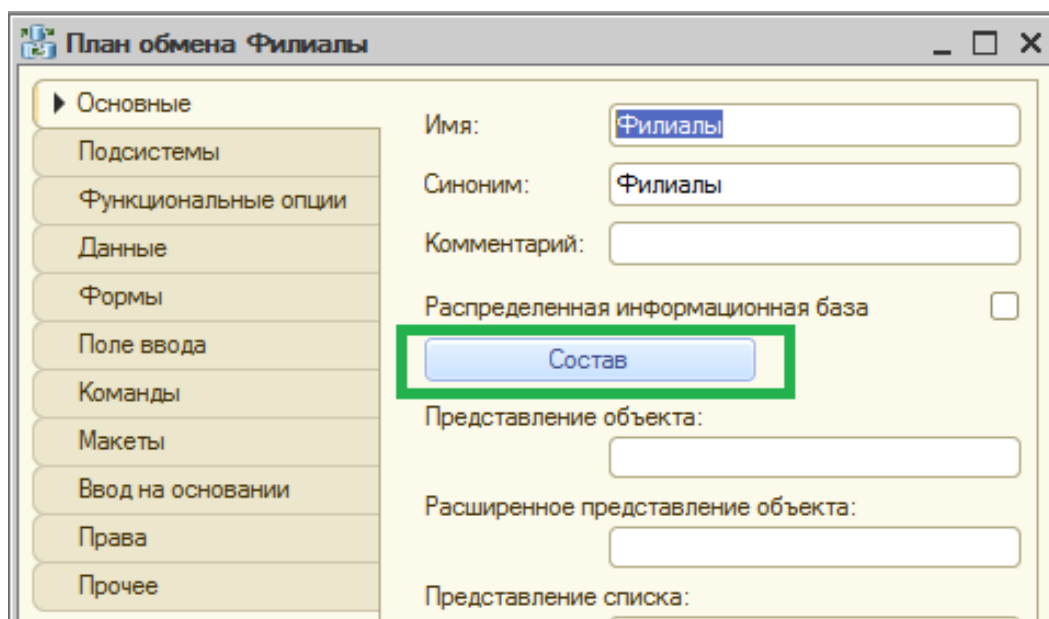


Рис. 8.4. Процедура-обработчик события ПриУстановкеНовогоКода

Константа ПрефиксНумерации не должна быть включена в состав обмена, поскольку ее значение должно быть уникальным для каждой базы, участвующей в обмене [1].

в) Создадим основную форму узла. В ней создадим обработчик события формы «При создании на сервере» (рис. 8.6). Он нужен для того, чтобы запретить установку реквизита Главный для предопределенного узла, соответствующего данной информационной базе [1].

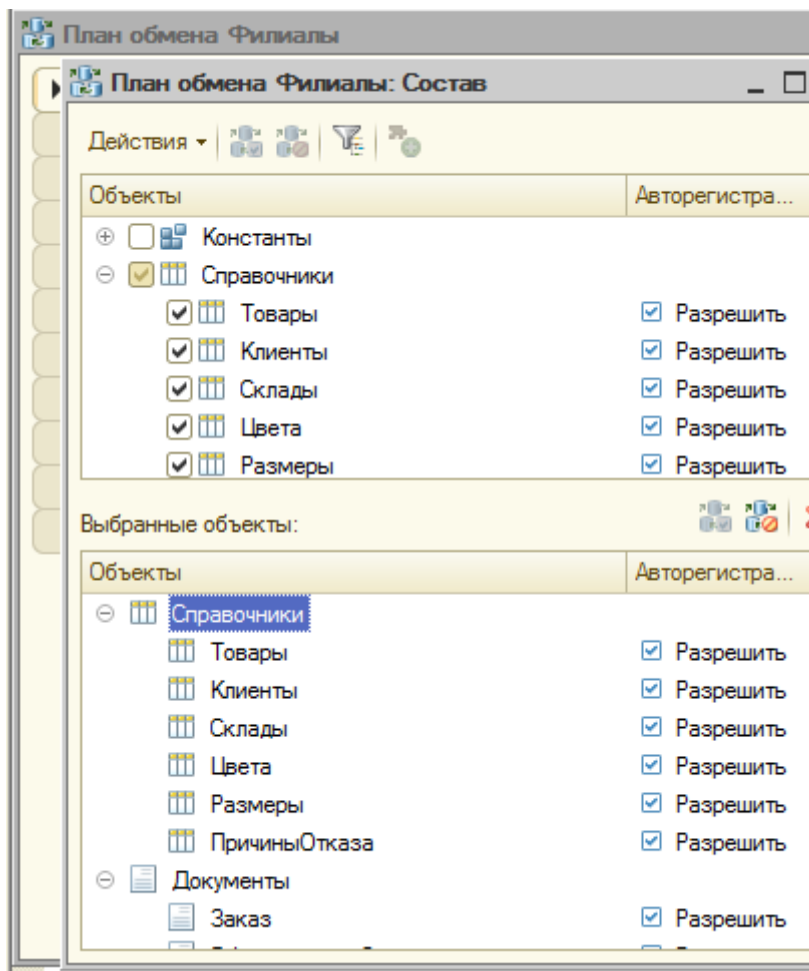


Рис. 8.5. Настройка состава объектов в плане обмена

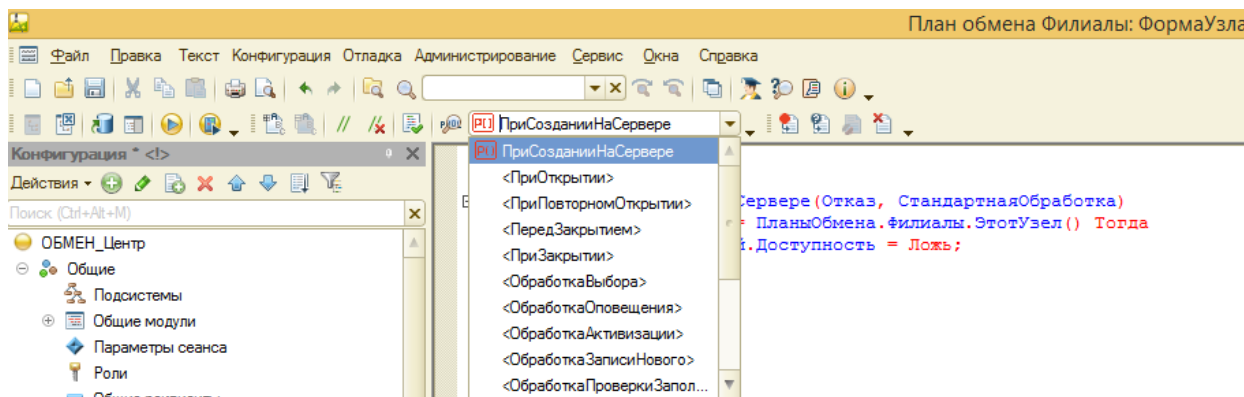


Рис. 8.6. Обработчик события формы «При создании на сервере»

Текст процедуры обработчика следующий (рис. 8.7):

```

&НаСервере
Процедура ПриСозданииНаСервере (Отказ, СтандартнаяОбработка)
    Если Объект.Ссылка = ПланыОбмена.филиалы.ЭтотУзел() Тогда
        Элементы.Главный.Доступность = Ложь;
    КонецЕсли;
КонецПроцедуры
    
```

Рис. 8.7. Обработчик события формы «При создании на сервере»

Метод менеджера плана обмена ЭтотУзел() возвращает ссылку на узел плана обмена, который соответствует данной информационной базе.

г) Создадим основную форму списка плана обмена [1].

При регистрации нового узла обмена мы должны будем сформировать для него все необходимые записи регистрации изменений для всех объектов конфигурации, входящих в данный план обмена. Это начальная синхронизация узла обмена всеми данными обмена.

На закладке Команды формы списка создадим команду ЗарегистрироватьИзменения. Вытащим ее на форму. При перетаскивании имя кнопки может не совпадать с именем команды. Отредактировать имя, чтобы название было «ЗарегистрироватьИзменения», т.к. в дальнейшем именно по такому имени мы будем обращаться к ней из встроеного языка [1].

Для команды создадим обработчик команды на клиенте и на сервере без контекста формы (рис. 8.8).

```

&НаСервереБезКонтекста
Процедура ЗарегистрироватьИзмененияНаСервере (Узел)
    // Регистрация изменений всех данных для узла.
    ПланыОбмена.ЗарегистрироватьИзменения (Узел) ;
КонецПроцедуры

&НаКлиенте
Процедура ЗарегистрироватьИзменения (Команда)
    ЗарегистрироватьИзмененияНаСервере (Элементы.Список.ТекущаяСтрока) ;
КонецПроцедуры

```

Рис. 8.8. Обработчик команды на клиенте и на сервере без контекста формы

В процедуру ЗарегистрироватьИзмененияНаСервере() нужно передавать в качестве параметра ссылку на объект ПланОбмена.Физикалы, используя свойство ТекущаяСтрока для таблицы Список [1].

В серверной процедуре обращаемся к механизму регистрации изменений, вызывая метод менеджера планов обмена – ЗарегистрироватьИзменения(), в который передается ссылка на текущий узел плана обмена. В результате выполнения этой процедуры в информационной базе будут созданы записи регистрации изменений, предназначенные для пересылки в созданный нами узел, для всех объектов обмена, указанных в составе данного плана обмена. Кнопка зарегистрировать изменения должны быть доступна только в случае, если текущий узел не является предопределенным для данной информационной базы, иначе регистрация изменений невозможна.

Для списка реализуем обработчик события ПриАктивизацииСтроки – на клиенте [1], рис. 8.9. В этой процедуре доступность кнопки зарегистрировать изменения определяется в зависимости от значения функции ПредопределенныйУзел(), в которую передается ссылка на текущий узел (Элемент.ТекущаяСтрока) [1], рис. 8.10.

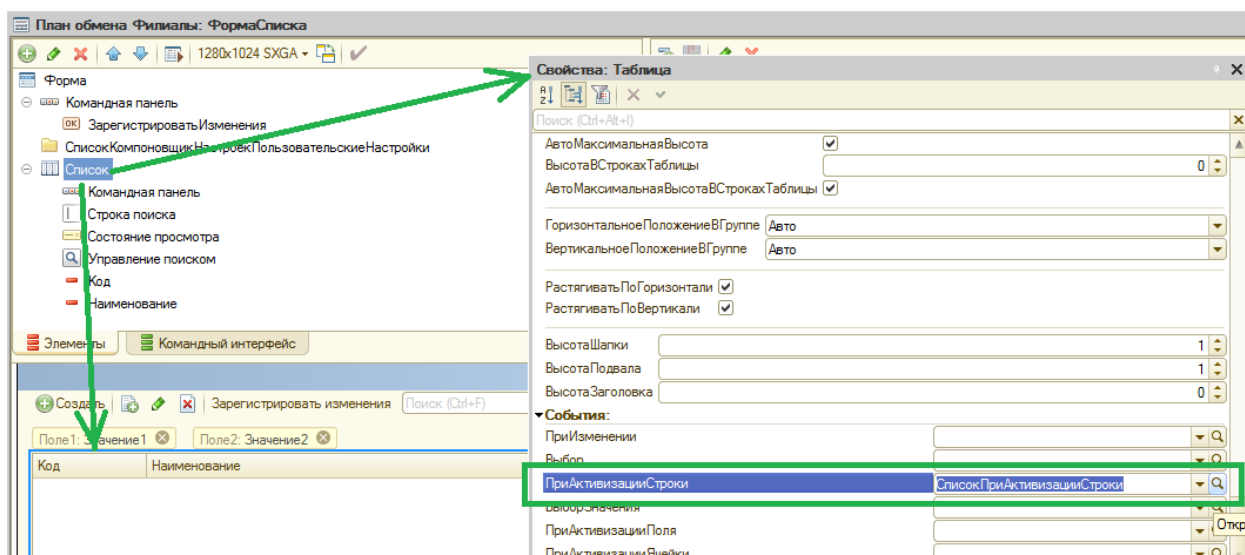


Рис. 8.9.

```

&НаСервереБезКонтекста
[ функция ПредопределенныйУзел (Узел)
    Возврат Узел = ПланыОбмена.филиалы.ЭтотУзел ();
    Конечфункции

&НаКлиенте
[ Процедура СписокПриАктивизацииСтроки (Элемент)
    Если ПредопределенныйУзел (Элемент.ТекущаяСтрока) Тогда
        Элементы.ЗарегистрироватьИзменения.Доступность = Ложь;
    Иначе
        Элементы.ЗарегистрироватьИзменения.Доступность = Истина;
    КонечЕсли;
    КонечПроцедуры

```

Рис. 8.10. Код обработчика события ПриАктивизацииСтроки

3) Создание процедур обмена данными.

а) Создадим обработку ОбменДанными. Создадим для нее форму обработки. На форме создадим команду ВыполнитьОбмен (обработчик типа на клиенте и на сервере без контекста) и вынесем ее на форму (рис. 8.11).

```

&НаКлиенте
[ Процедура ВыполнитьОбмен (Команда)
    ВыполнитьОбменНаСервере ();
    КонечПроцедуры

&НаСервереБезКонтекста
[ Процедура ВыполнитьОбменНаСервере ()
    ВыборкаУзлов = ПланыОбмена.филиалы.Выбрать ();
    Пока ВыборкаУзлов.Следующий () Цикл
        // Произвести обмен данными со всеми узлами, кроме текущего (ЭтотУзел).
        Если ВыборкаУзлов.Ссылка <> ПланыОбмена.филиалы.ЭтотУзел () Тогда
            УзелОбъект = ВыборкаУзлов.ПолучитьОбъект ();
            // Получить сообщение.
            УзелОбъект.ПрочитатьСообщениеСИзменениями ();
            // Сформировать сообщение.
            УзелОбъект.ЗаписатьСообщениеСИзменениями ();
        КонечЕсли;
    КонечЦикла;
    КонечПроцедуры

```

Рис. 8.11. Код обработчика команды ВыполнитьОбмен

В цикле перебираем узлы, которые содержатся в плане обмена Филиалы, и для всех узлов, кроме себя самого, производим сначала чтение сообщений, поступивших из других узлов обмена [1].

Затем мы формируем для них сообщения, предназначенные для передачи и содержащие измененные данные для этого узла.

б) Сами процедуры записи и чтения данных обмена разместим в модуле объекта план обмена Филиалы.

Процедура записи данных обмена приведена на рис. 8.12 [1].

```

Процедура ЗаписатьСообщениеСИзменениями() Экспорт
// задание в качестве каталога обмена - каталога временных файлов|
Каталог=КаталогВременныхФайлов();

Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = "----- Выгрузка в узел " + Строка(ЭтотОбъект) + " -----";
Сообщение.Сообщить();

// Сформировать имя временного файла.
Имяфайла = Каталог +?(Прав(Каталог, 1) = "\", "\", "\"+"Message" + СокрЛП(ПланыОбмена.Филиалы.ЭтотУзел().Код) +
"_" + СокрЛП(Ссылка.Код) + ".xml";

// Создать объект записи XML
// *** ЗаписьXML-документов.
ЗаписьXML = Новый ЗаписьXML;
ЗаписьXML.ОткрытьФайл(Имяфайла);
ЗаписьXML.ЗаписатьОбъявлениеXML();

// *** Инфраструктура сообщений.
ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();
ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Ссылка);
Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = " Номер сообщения: " + ЗаписьСообщения.НомерСообщения;
Сообщение.Сообщить();

// Получить выборку измененных данных
// *** Механизм регистрации изменений.
ВыборкаИзменений = ПланыОбмена.ВыбратьИзменения(ЗаписьСообщения.Получатель, ЗаписьСообщения.НомерСообщения);
Пока ВыборкаИзменений.Следующий() Цикл
    Данные = ВыборкаИзменений.Получить();

    // Записать данные в сообщение *** XML-сериализация.
    ЗаписатьXML(ЗаписьXML, Данные);
КонечЦикла;

ЗаписьСообщения.ЗакончитьЗапись();
ЗаписьXML.Закреть();

Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = "----- Конец выгрузки -----";
Сообщение.Сообщить();
КонецПроцедуры

```

Рис. 8.12. Код процедуры записи данных обмена

Комментарии к коду:

– упрощенно будем обмениваться сообщениями через каталог временных файлов.

– сообщаем пользователю о начале выгрузки.

– формируем имя файла, который будет содержать данные для обмена.

Имена сообщений стандартизованы и имеют вид *MessageКодУзлаОтправителя_Получателя.xml*.

– обращаемся к механизмам записи/чтения XML-документов и создадим новый объект *ЗаписьXML*. С помощью него откроем новый XML-файл для записи, запишем в него объявление XML.

Если формируется XML-документ, а не фрагмент XML-документа, то в начало документа надо поместить его объявление. Для того чтобы поместить в документ его объявление, необходимо вызвать данный метод. При этом в начало документа помещается текст следующего вида: `<?xml version="1.0"?>`. Если при

открытии файла или установке строки была указана кодировка, то в объявление также помещается и атрибут 'encoding'. Например: `<?xml version="1.0" encoding="UTF-8"?>`. Запись объявления XML должна выполняться перед другими операциями записи XML [1].

– теперь обратимся к механизмам инфраструктуры сообщений и создадим новый объект *ЗаписьСообщенияОбмена*. Метод этого объекта *НачатьЗапись()* позволяет создать очередной номер сообщения и записать заголовок сообщения в XML. Поскольку мы находимся в модуле объекта, то мы используем стандартный реквизит *Ссылка* в качестве ссылки на объект план обмена *Филиалы*.

– далее, чтобы получить данные, которые необходимо сохранить в этом файле, мы обратимся к механизму регистрации изменений и получим выборку из записей регистрации изменений, предназначенных данному узлу. При формировании выборки мы передаем вторым параметром номер сообщения [1].

– перебираем выборку записей в цикле и сериализуем их в открытый XML-файл.

– закончим запись сообщения, запишем запись XML и закроем файл.

– сообщаем пользователю об окончании выгрузки данных в узел.

Процедура чтения данных обмена (рис. 8.13).

```

Процедура ПрочитатьСообщениеСИзменениями() Экспорт
    Каталог=КаталогВременныхФайлов();
    // Сформировать имя файла.
    ИмяФайла = Каталог + ?(Прав(Каталог, 1) = "\", "", "\")+ "Message" + СокрЛП(Ссылка.Код)
        + "_" +СокрЛП(ПланыОбмена.Филиалы.ЭтотУзел().Код) + ".xml";

    файл = Новый файл(ИмяФайла);
    Если Не файл.Существует() Тогда
        Возврат;
    КонецЕсли;

    // *** Чтение документов XML
    // Попытаться открыть файл.
    ЧтениеXML = Новый ЧтениеXML;
    Попытка
        ЧтениеXML.ОткрытьФайл(ИмяФайла);
    Исключение
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = "Невозможно открыть файл обмена данными.";
        Сообщение.Сообщить();
        Возврат;
    КонецПопытки;

    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "----- Загрузка из " + Строка(ЭтотОбъект) + " -----";
    Сообщение.Сообщить();
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = " - Считывается файл " + ИмяФайла;
    Сообщение.Сообщить();
    // Загрузить из найденного файла
    // *** Инфраструктура сообщений.
    ЧтениеСообщения = ПланыОбмена.СоздатьЧтениеСообщения();
    // Читать заголовок сообщения обмена данными - файла XML.
    ЧтениеСообщения.НачатьЧтение(ЧтениеXML);

    // Сообщение предназначено не для этого узла.
    Если ЧтениеСообщения.Отправитель <> Ссылка Тогда
        ВызватьИсключение "Неверный узел";
    КонецЕсли;

    // Удаляем регистрацию изменений для узла отправителя сообщения.
    // *** Служба регистрации изменений.
    ПланыОбмена.УдалитьРегистрациюИзменений(ЧтениеСообщения.Отправитель, ЧтениеСообщения.НомерПринятого);

```

Рис. 8.13. Код процедуры чтения данных обмена

```

// Читаем данные из сообщения *** XML-сериализация.
Пока ВозможностьЧтенияXML(ЧтениеXML) Цикл
    // Читаем очередное значение.
    Данные = ПрочитатьXML(ЧтениеXML);
    // Не переносим изменение, полученное в главный из неглавного, если есть регистрация изменения.
    Если Не ЧтениеСообщения.Отправитель.Главный И
        ПланыОбмена.ИзменениеЗарегистрировано(ЧтениеСообщения.Отправитель, Данные) Тогда
        Сообщение = Новый СообщениеПользователю;
        Сообщение.Текст = " - Изменения отклонены";
        Сообщение.Сообщить();
        Продолжить;
    КонецЕсли;
    // Записать полученные данные.
    Данные.ОбменДанными.Отправитель = ЧтениеСообщения.Отправитель;
    Данные.ОбменДанными.Загрузка = Истина;
    Данные.Записать();
КонецЦикла;

ЧтениеСообщения.ЗакончитьЧтение();
ЧтениеXML.Закрыть();
УдалитьФайлы(ИмяФайла);

Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = "----- Конец загрузки -----";
Сообщение.Сообщить();

КонецПроцедуры

```

Рис. 8.13. Код процедуры записи данных обмена (продолжение)

Комментарии к коду [1]:

– упрощенно будем обмениваться сообщениями через каталог временных файлов.

– сообщаем пользователю о начале процедуры чтения.

– формируем имя файла, который будет содержать данные для обмена.

Имена сообщений стандартизованы и имеют вид *MessageКодУзлаОтправителя_Получателя.xml*. Формируем имя файла, которое надеемся найти в этом каталоге, а затем создав новый объект Файл с таким именем, проверяем, существует ли он. Если такого файла нет, мы завершаем работу процедуры.

– команды чтения найденного файла с данными обмена: обращаемся к механизмам записи/чтения документов XML, которые работают с ними на базовом уровне. Создаем новый объект Чтение XML, с помощью которого открываем найденный файл для чтения. В случае успеха выводим сообщение о начале загрузки из файла.

– обращаемся к механизмам инфраструктуры сообщений планов обмена и создаем объект ЧтениеСообщенияОбмена. Используя метод этого объекта НачатьЧтение() мы считываем заголовок XML-сообщения, в котором содержится в том числе информация об отправителе сообщения.

– когда мы представили данные обмена в виде сообщения и получили его заголовок, можно произвести проверку, перед тем как начать собственно обрабатывать данные. Проверяем, является ли отправитель сообщения тем узлом, для которого мы в данном вызове этой процедуры производим обмен данными.

– если все в порядке, то перед тем, как начать чтение данных, следует удалить все записи регистрации изменений, которые были сделаны для этого узла и соответствовали номерам сообщений меньше или равным указанному в обрабатываемом нами сообщении как номер принятого. Это делается для того,

чтобы исключить дублирование данных, которые уже ранее были посланы этому узлу и им обработаны. Для исполнения этих действий мы обращаемся к службе регистрации изменений и используем метод *УдалитьРегистрациюИзменений()*.

– далее приступаем к чтению непосредственно самих данных, содержащихся в сообщении. Чтение данных выполняется в цикле, причем мы снова обращаемся к механизмам *XML*-сериализации и методом глобального контекста *ВозможностьЧтенияXML()* получаем очередной тип данных *XML* из объекта *ЧтениеXML* и определяем, имеется ли соответствующий тип «1С:Предприятия». В случае неуспеха выполнение цикла прекращается [2].

– далее внутри цикла: нам необходимо представить данные *XML* в виде некоторого значения, имеющего тип «1С:Предприятия». Для этого мы используем метод глобального контекста *ПрочитатьXML()*. В результате выполнения этого метода переменная *Данные* будет содержать объект «1С:Предприятие», соответствующий данным *XML*.

– теперь после того как объект «1С:Предприятие» получен, следует разрешить возможную коллизию. Проверяем, является ли узел-отправитель главным узлом и есть ли запись об изменении этого объекта для главного узла в нашей базе данных. Если объект изменялся в нашей базе и отправитель не является главным узлом, то мы отклоняем запись полученного объекта. Во всех остальных случаях мы принимаем изменения полученного объекта.

– записываем полученные данные. Перед записью мы устанавливаем у него в параметрах обмена данными узел отправителя для того, чтобы система при записи этого объекта в нашей базе данных не формировала записи регистрации изменений этого объекта для того узла, от которого мы его только что получили. Кроме того, в параметрах обмена данными мы устанавливаем свойство загрузка, информирующее систему о том, что запись объекта будет происходить в режиме обновления данных, полученных в результате обмена. Такое указание позволяет системе упростить процедуру записи объекта, отказавшись от ряда стационарных проверок и исключив изменения связанных данных, которые выполняются при обычной записи.

– после того как все сообщение будет нами обработано, заканчиваем чтение.

– прекращаем чтение *XML*-данных из файла методом *Закреть()*.

– после того как все данные, содержащиеся в файле, обработаны, файл можно удалить [2].

4) Проверка работы обмена данными.

Для проверки работоспособности механизма необходимо сохранить полученную конфигурацию в файл. Создать еще одну базу, куда загрузить выгрузку конфигурации. Переименовать в свойствах конфигурации: у первой конфигурации задать Имя равное «Обмен_Центр», а у второй – «Обмен_Филиал» [2].

а) Запустить конфигурацию «Обмен_Центр» в режим исполнения.

Задать значение константы ПрефиксНумерации – СВ

Открыть план обмена Филиалы. В нем уже присутствует одна предопределенная запись. Необходимо задать параметры узла по умолчанию, то есть параметры самой базы «Обмен_Центр». Отредактируем эту запись.

Код – СВ, наименование – Central database.

Создадим еще один узел, который будет соответствовать базе филиала.

Код – ВВ, наименование – Branch base.

Именно код идентифицирует узла обмена в различных базах, поэтому в базе филиала мы будем создавать узлы с такими же кодами.

Выделим в списке новый узел Branch base и нажмем кнопку *Зарегистрировать изменения*.

Далее вызовем обработку Обмен данными и нажмем *Выполнить обмен*.

В результате обмена данными центральная база сформировала файл обмена, содержащий изменения всех данных, которыми она обменивается с филиалом [2].

б) Запустить конфигурацию «Обмен_Филиал» в режиме конфигуратора.

Для тех справочников и других объектов, которые имели предопределенные элементы, необходимо на закладке Прочее в свойствах объекта установить свойство *Обновление предопределенных данных* в значение *Не обновлять автоматически*.

Запустить в режим исполнения.

Задать значение константы ПрефиксНумерации – ВВ

Открыть план обмена и описать предопределенный узел, соответствующий базе филиала. Код – ВВ, наименование – Branch base.

Создадим еще один узел, который будет соответствовать базе центрального офиса.

Код – СВ, наименование – Central database. Установим признак Главный. Запишем элемент.

Выделим в списке узлов обмена новый узел Центральная база и нажмем кнопку *Зарегистрировать изменения*.

Запустим обработку *Обмен данными* и нажмем *Выполнить обмен*.

Справочники и другие объекты конфигурации, участвующие в обмене данными, будут заполнены элементами.

Обмен данными от центральной базы в базу филиала осуществлен.

в) Проверка обмена в другую сторону. Создадим в каком-нибудь справочнике новый элемент. Обратим внимание, что для нового элемента код будет сформирован с префиксом ВВ. После этого снова нажмем *Выполнить обмен* в открытой форме обработки *Обмен данными* в базе «Обмен_Филиал».

Затем перейдем в центральную базу, также выполним обмен и убедимся, что элемент справочника, созданный в базе филиала, перенесен в центральную базу [1].

Механизм распределенных информационных баз. Этот механизм является развитием универсального механизма обмена данными.

Предназначен для обмена данными только с идентичными конфигурациями «1С:Предприятия» и жестко регламентирует структуру создаваемой системы.

Распределенная информационная база – это совокупность информационных баз «1С:Предприятия» (узлов распределенной информационной базы), в которых поддерживается синхронизация конфигурации и данных. Распределенная информационная база имеет иерархическую структуру. У каждого узла распределенной информационной базы может быть один главный и произвольное число подчиненных узлов. «Самый главный узел» или узел, у которого нет главного узла, называется корневым узлом распределенной информационной базы. Каждый из узлов может обмениваться данными только со своими «соседями», то есть со своим главным и подчиненными узлами.

Механизм распределённых информационных баз реализуется планами обмена. Для этого объект конфигурации План обмена содержит свойство **Распределенная информационная база**.

Если свойство установлено, то для данного плана обмена включается механизм распределённых информационных баз и разработчик получает возможность создать распределённую базу исключительно интерактивными средствами, без написания кода. Такая возможность не исключает программного управления обменом, которое также доступно при работе с распределёнными информационными базами.

Распределенная информационная база должна иметь четко определенную древовидную структуру (рис. 8.14). Количество уровней в такой структуре не ограничено, главное – между двумя связанными узлами всегда должно быть определено отношение «главный-подчиненный».

Т.о., любой узел этой структуры может иметь произвольное количество подчиненных узлов (в том числе и ни одного). Кроме этого, все узлы, кроме одного, должны иметь по одному главному узлу. И один узел не будет иметь главного узла – это корневой узел. Такое жесткое задание структуры узлов необходимо для определения порядка миграции изменений данных и изменений конфигурации.

Конфигурация может быть изменена только в корневом узле. Изменения данных могут передаваться между любыми узлами.

Разрешение коллизий также будет производиться исходя из отношения «главный-подчиненный». Если изменения выполнены одновременно и в главном, и в подчиненном узле, при обмене данными будут приняты только изменения главного узла, а изменения подчиненного отвергнуты.

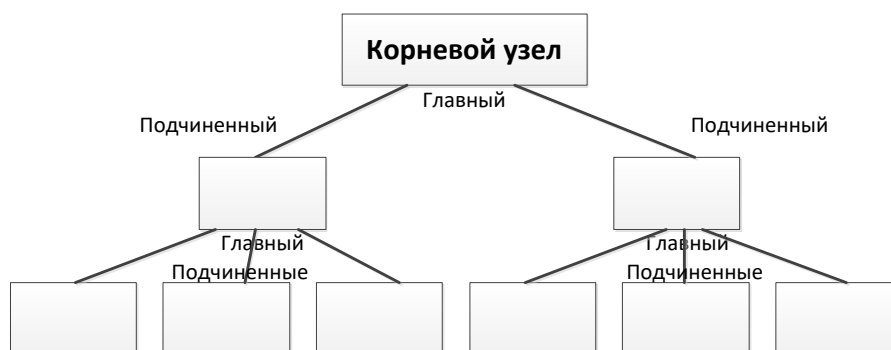


Рис. 8.14. Механизм распределённых информационных баз

Тема 9. Механизм «1С:Аналитика»

Обзор возможностей механизма «1С:Аналитика». Общие приемы работы.

«1С:Аналитика» – это новый компонент, который появился в составе платформы «1С:Предприятие» для решения задач оперативного просмотра и анализа данных [4].

«1С:Аналитика» работает как составная часть платформы и предоставляет пользователям прикладных решений «1С:Предприятия 8» дополнительный визуальный интерфейс, позволяющий быстро и легко получать нужную информацию и анализировать ее с разных сторон. Рабочий стол «1С:Аналитика» представлен на рис. 9.1.

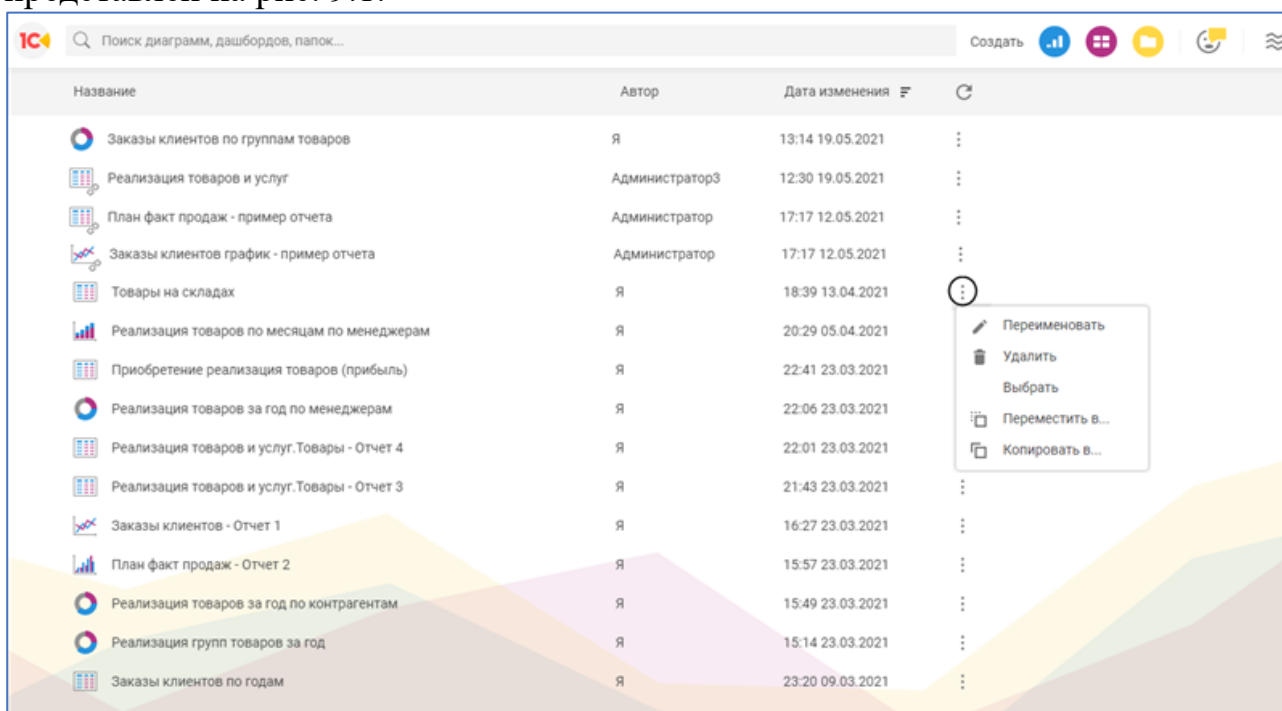


Рис. 9.1. Рабочий стол «1С:Аналитика»

«1С:Аналитика» обладает рядом уникальных особенностей и преимуществ перед другими аналитическими системами [4]:

– «1С:Аналитика» – это часть платформы «1С:Предприятие», работающая в рамках настроенных в прикладном решении прав доступа к данным. То есть пользователь «1С:Аналитики» имеет доступ ровно к тем данным прикладного решения, на которые у него есть права. Кроме того, он работает в «1С:Аналитике» с той же учетной записью, что и в прикладном решении, и не нуждается в повторной авторизации. Также пользователь «1С:Аналитики» может анализировать информацию в том виде (в тех структурах данных), в котором она хранится в прикладном решении.

– Интерфейс «1С:Аналитики» максимально прост и удобен для бизнес-пользователей, которые не обладают специальными техническими навыками и, возможно, не имеют опыта работы с «1С:Предприятием».

– Не нужно беспокоиться о выгрузке данных из «1С:Предприятия» и их загрузке в аналитическую систему. «1С:Аналитика» работает с данными непосредственно в прикладном решении, причем позволяет быстро их обрабатывать, не замедляя работу остальных пользователей. Пользователь «1С:Аналитики» анализирует и видит непосредственно те же данные, что лежат в прикладном решении, без промежуточных программ преобразования и выгрузки данных.

– В «1С:Аналитике» можно быстро и легко получить сводную информацию по всем записям в нужном регистре или по документам, а потом детализировать эти данные до отдельного документа или элемента справочника. Из отчета в «1С:Аналитике» легко получить ссылку на нужный объект данных в «1С:Предприятии» и открыть его там для изменения или дальнейшего изучения.

– «1С:Аналитика» – часть мира «1С», поэтому изначально поддерживает иерархию элементов в справочниках и ссылки на объекты данных.

Общая схемы работы в «1С:Аналитике» состоит из следующих этапов [4]:

1) Выбор источника данных для диаграммы.

Можно построить диаграмму на базе различных объектов конфигурации (регистров накопления, регистров сведений, справочников, документов или их табличных частей и др.). В качестве основы для построения диаграммы также можно использовать дополнительные источники данных, которые создаются на языке запросов и объединяют информацию из разных объектов прикладного решения.

2) Добавление измерений, фактов и фильтров.

После выбора источника данных вы можете наполнять свою диаграмму требующимися для анализа измерениями, фактами, содержащимися в источнике данных, а также добавлять фильтры, чтобы в диаграмму попали только нужные данные.

Сервер «1С:Аналитики» запросит из источника все данные в соответствии с наложенными фильтрами, а затем сгруппирует факты по выбранным измерениям.

Группировка факта делается агрегирующей функцией, установленной на поле факта: суммой, количеством записей, средним и др.

3) Выбор вида диаграммы и представления данных.

После того как нужные данные получены, вы можете менять форму их представления, выбирать представление диаграммы в виде таблицы, графика и т. п.

4) Детализация данных.

Вы можете быстро и легко построить диаграмму, показывающую в табличном виде «верхний» срез данных. Например, показать заказы клиентов по годам. А затем уже для выделения и анализа конкретного фрагмента информации добавлять дополнительные фильтры и делать отборы тех данных, которые вас интересуют. И потом при необходимости «раскрывать» полученные данные до отдельных первичных документов «1С:Предприятия».

Это соответствует общему принципу анализа данных [4]:

Сначала сделать общий анализ информации по ключевым показателям компании.

Если ключевые показатели отличаются от целевых значений, детализировать эти данные на участках, где идет отклонение, вплоть до выяснения первопричины таких отклонений.

5) Публикация диаграммы.

После создания диаграммы вы можете опубликовать свою диаграмму в системе и передать ссылку на нее другим пользователям или руководству компании.

б) Создание дашборда.

Для удобства сводного представления данных и показа нескольких связанных друг с другом диаграмм вы можете создать дашборд и вывести на него все необходимые диаграммы для комплексного представления данных по нужной тематике.

Для каждой диаграммы или дашборда на рабочем столе показываются название и дата последнего изменения, имя пользователя, создавшего диаграмму, а также слева от названия выводится иконка, отображающая тип диаграммы (таблица, график, столбчатая диаграмма и т. п.).

Нажав на диаграмму, пользователь может просмотреть ее или отредактировать.

Нажав на три вертикальные точки справа от даты изменения диаграммы, можно вызвать контекстное меню, содержащее команды для переименования, копирования, перемещения и удаления диаграммы.

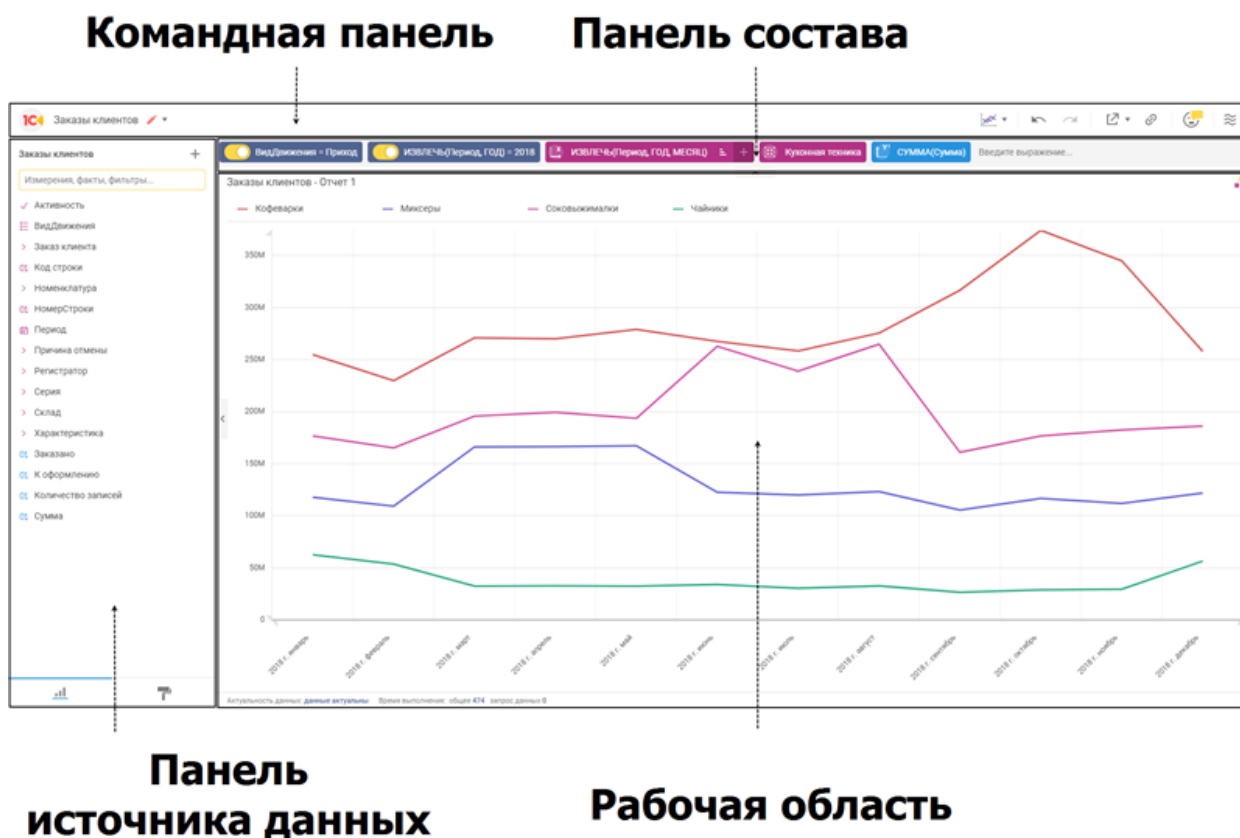


Рис. 9.2. Панель источника данных и панель состава [4]

Переименовывать и удалять можно только собственные диаграммы, автором которых вы являетесь. Диаграммы других пользователей вы можете только открепить, то есть удалить со своего рабочего стола, но не с сервера «1С:Аналитики».

Копировать диаграмму можно в том числе и в созданную ранее папку.

В верхней части рабочего стола расположены строка поиска диаграмм и кнопки создания диаграммы, дашборда и папки.

Редактор диаграммы является основным рабочим местом пользователя в «1С:Аналитике». В режиме просмотра диаграммы отображаются командная панель и рабочая область. В режиме редактирования на экране появляются также панель источника данных и панель состава (рис. 9.2).

Командная панель. В командной панели собраны функции, позволяющие переименовать диаграмму, отменить последние изменения, опубликовать диаграмму, получить на нее ссылку, открыть системное меню и т. д. (рис. 9.3).

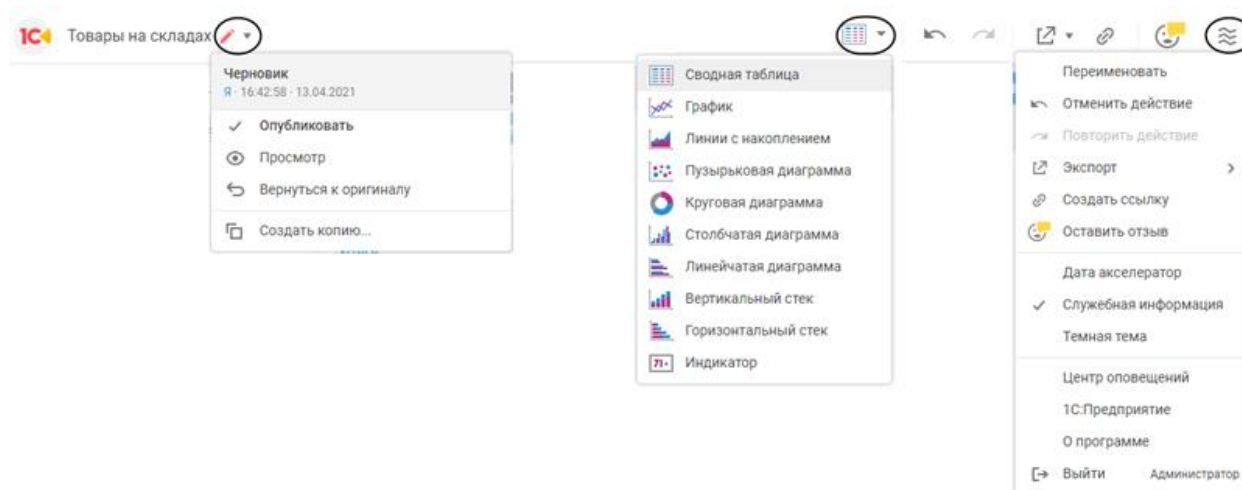


Рис. 9.3. Командная панель диаграммы

Панель источника данных. Панель источника данных содержит все исходные поля (измерения, факты и сохраненные фильтры), которые вы можете использовать для наполнения диаграммы данными.

Измерения – это разрезы, в которых вы хотите анализировать информацию (например, периоды, номенклатура, контрагенты), а факты – это числовые значения, которые будут агрегированы (например, получена сумма продаж) в этих аналитических разрезах.

Состав полей в источнике соответствует исходным данным объекта из информационной базы «1С:Предприятия» [4].

Например, если в качестве источника для диаграммы выбрать регистр накопления Заказы клиентов, то стандартные реквизиты, измерения и реквизиты регистра будут добавлены на панель источника данных «1С:Аналитики» в качестве измерений, а ресурсы – в качестве фактов

Для простоты и удобства добавление поля в состав диаграммы производится одинарным щелчком мыши (или касанием для мобильных устройств) на нем в панели источника (рис. 9.3).

В «1С:Аналитике» все поля имеют две роли: *измерение* и *факт*. В состав диаграммы также входят фильтры.

Фильтры определяют, какие данные из информационной базы попадут в диаграмму. Затем все выбранные факты группируются по входящим в состав отчета измерениям. Измерения в «1С:Аналитике» отмечены пурпурным цветом ■, факты – сине-голубым ■, а фильтры – серо-синим ■.

Состав полей в источнике соответствует исходным данным объекта из информационной базы «1С:Предприятия». Например, если в качестве источника для диаграммы выбрать регистр накопления *Заказы клиентов*, то можно увидеть, что «1С:Аналитика» предлагает поля, соответствующие данным регистра (измерениям, ресурсам, реквизитам, стандартным реквизитам) в конфигураторе «1С:Предприятия» [4].

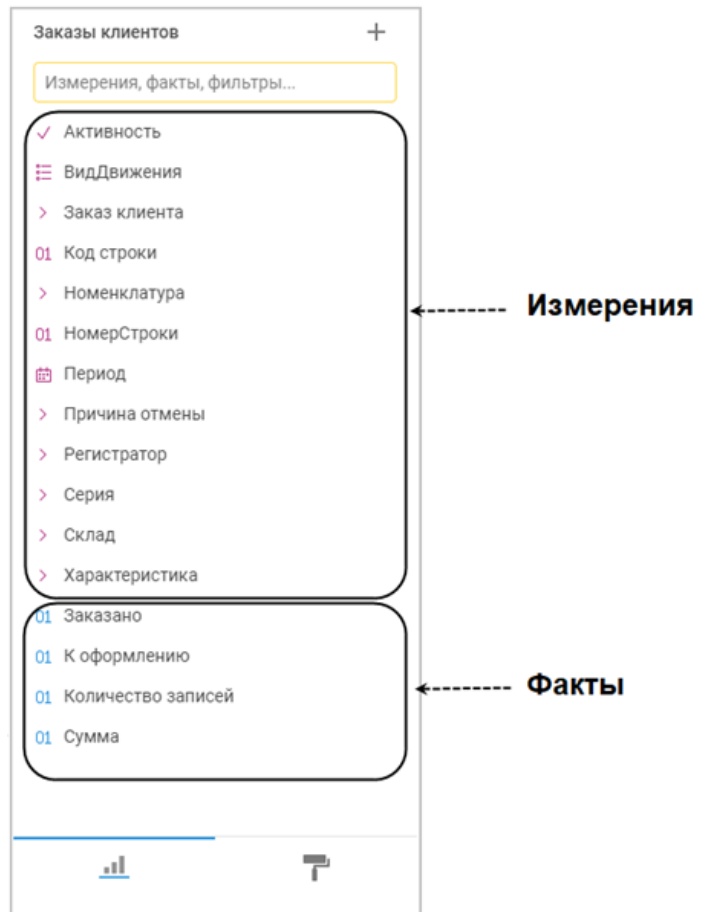


Рис. 9.3. Панель источника данных

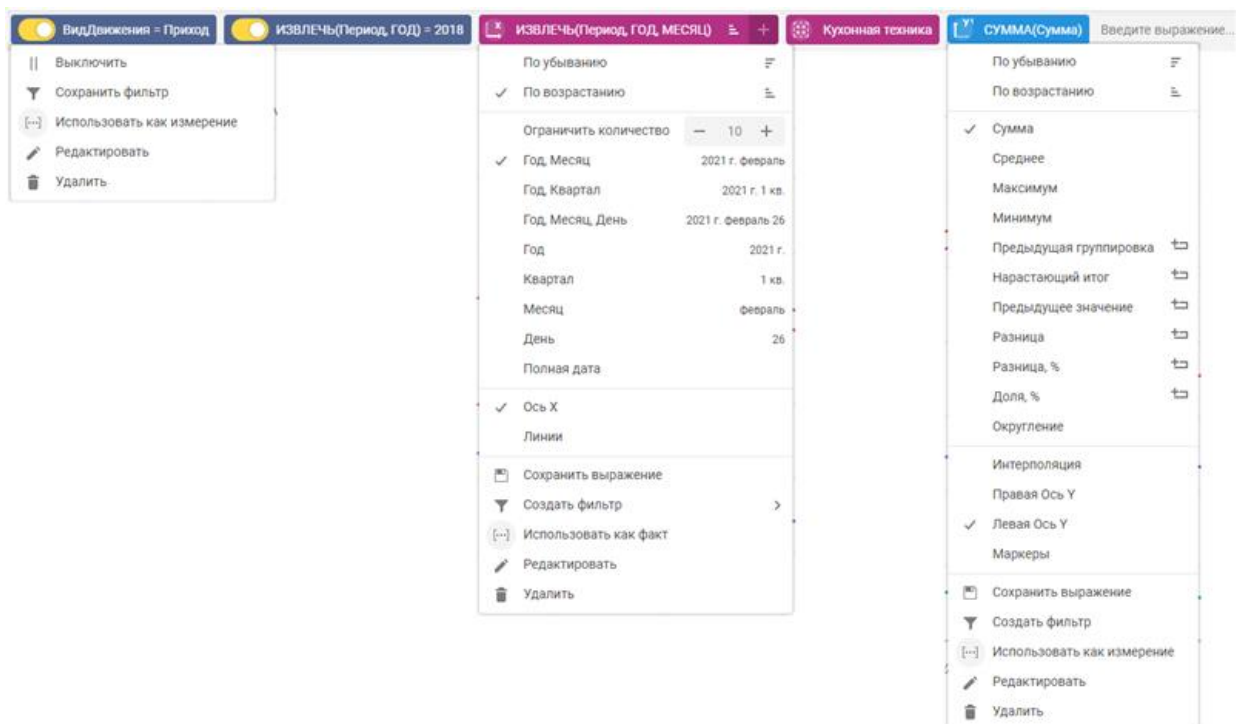


Рис. 9.3. Панель состава диаграммы

Панель состава. Панель состава диаграммы содержит все измерения, факты, которые вы добавили в нее из источника данных, а также фильтры, которые вы создали, чтобы ограничить показ данных в диаграмме (рис. 9.4). Также в панели состава диаграммы могут присутствовать и собственные поля.

Каждое поле в составе диаграммы имеет свое контекстное меню, которое открывается при нажатии на поле [4]. Это меню содержит как общие функции, так и функции, специфичные для роли поля.

Рабочая область. В рабочей области пользователь видит результат построения диаграммы. Результат меняется «на лету», при каждом изменении состава полей диаграммы и их настроек, а также при изменении графического представления диаграммы [4].

Так, продажи товаров в разрезе менеджеров могут быть показаны в виде круговой диаграммы (рис. 9.4). А при изменении вида диаграммы на столбчатую изображение сразу же изменится соответствующим образом (рис. 9.5).

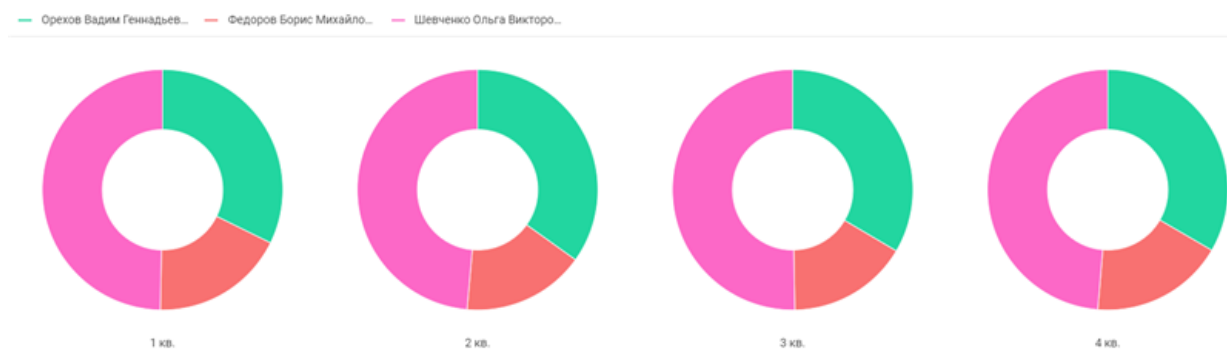


Рис. 9.4. Продажи товаров по менеджерам в виде круговой диаграммы

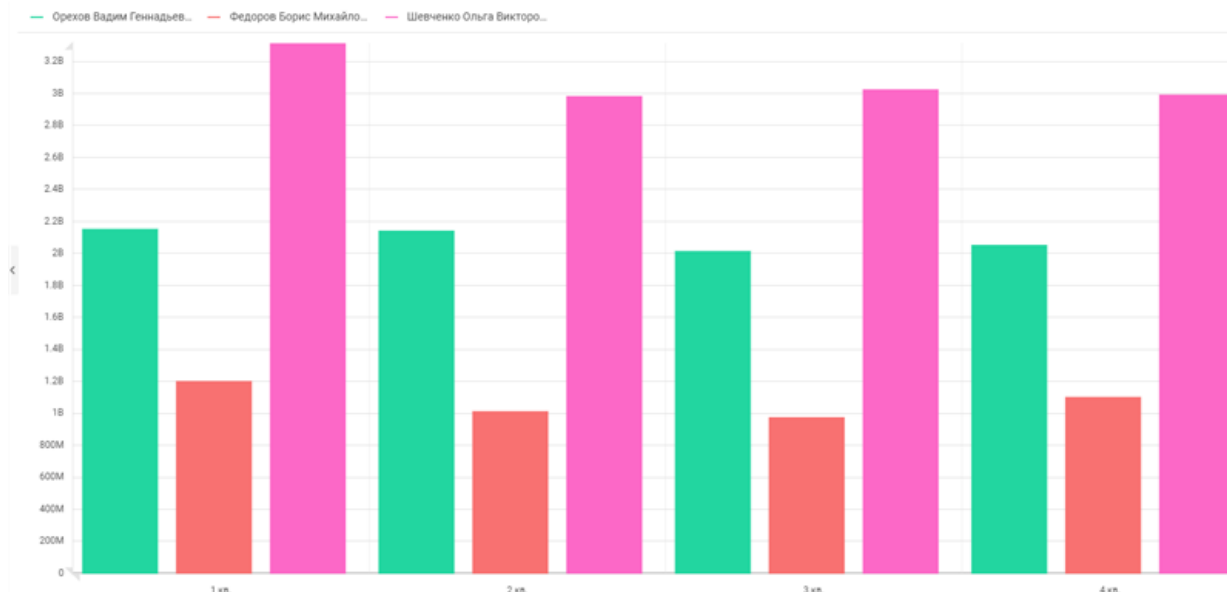


Рис. 9.5. Продажи товаров по менеджерам в виде столбчатой диаграммы

Группировка данных

Часто при анализе данных хочется объединить какие-то позиции в аналитическом разрезе и посмотреть их как единое целое. Например, построив диаграмму продаж по номенклатуре, хочется объединить ряд товаров в одну группу и посмотреть данные по этой группе.

«1С:Аналитика» позволяет выполнить эту операцию непосредственно при разработке диаграммы с использованием функции построения группировки.

Например, вам нужно проанализировать сезонные тренды продаж товаров. Если построить диаграмму продаж непосредственно по элементам номенклатуры, то, как видно на рис. 9.6, большое количество элементов не позволяет увидеть тренды продаж.

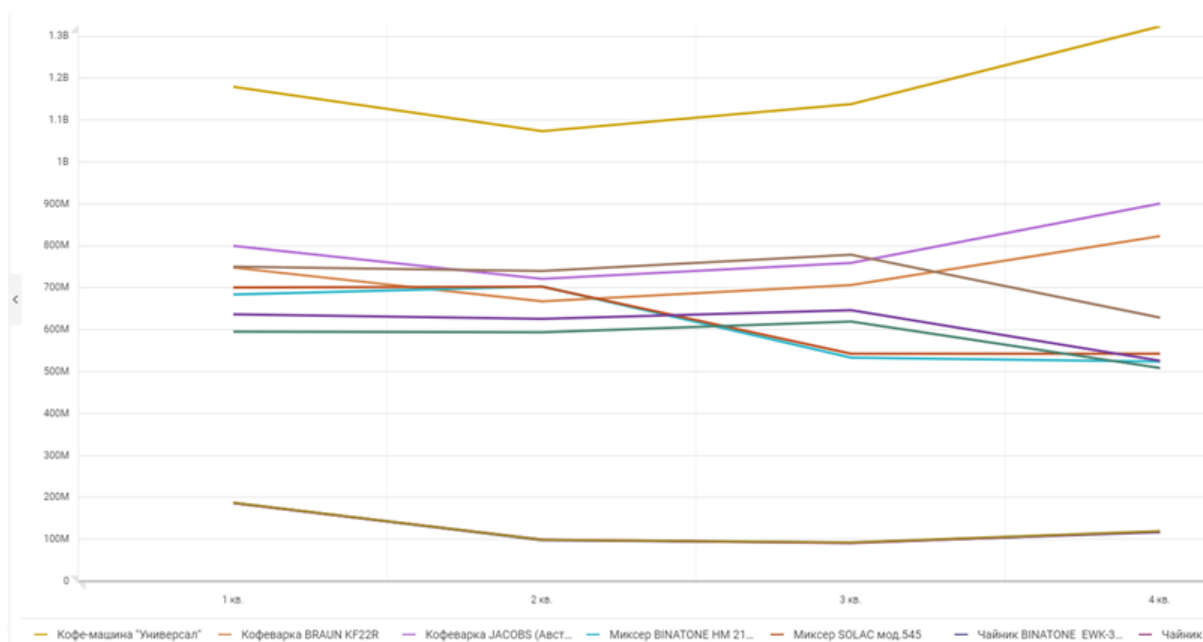


Рис. 9.6. Продажи товаров без группировки данных

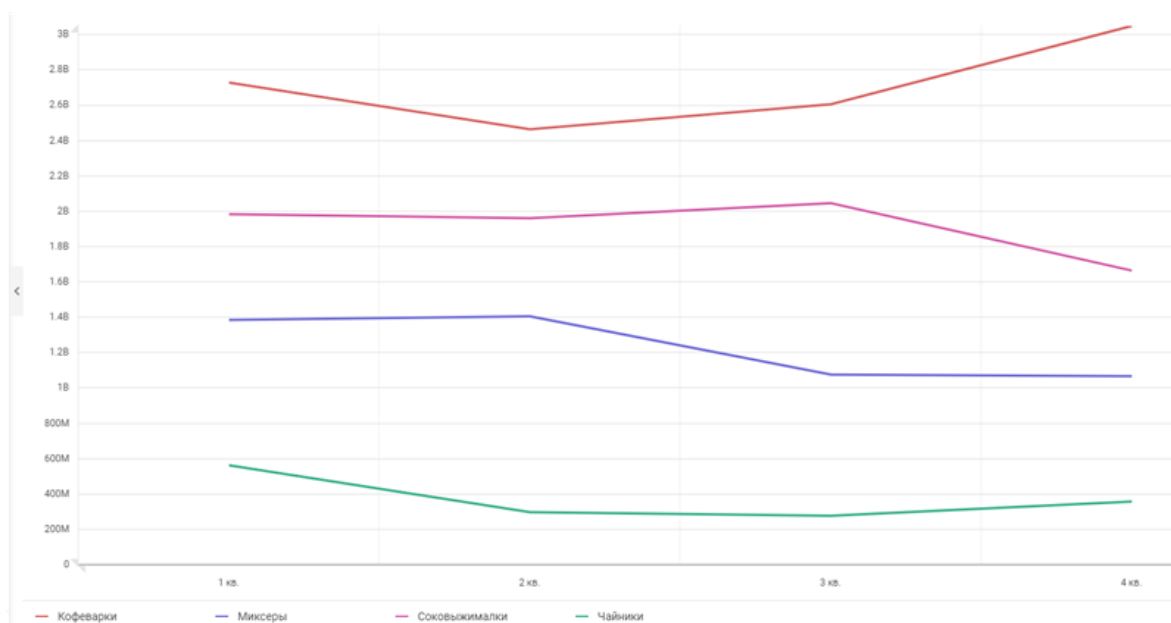


Рис. 9.7. Продажи товаров с группировкой данных

В то время как эти же данные, но с группировкой по типам товаров, на рис. 9.7 явно демонстрируют сезонность продаж для каждой из групп товаров.

Графическое представление диаграмм

После того как вы определились с набором полей, фильтров и группировок диаграммы, вы можете выбрать графическое представление, которое, на ваш взгляд, лучше и нагляднее показывает данные диаграммы. Для этого нужно нажать соответствующую кнопку в командной панели и выбрать желаемый вид диаграммы (рис. 9.8). Данные могут быть представлены в виде [4]:

- сводной таблицы;
- графика;
- линий с накоплением;
- пузырьковой диаграммы;
- круговой диаграммы;
- столбчатой диаграммы;
- линейчатой диаграммы;
- вертикального стека;
- горизонтального стека;
- индикатора.

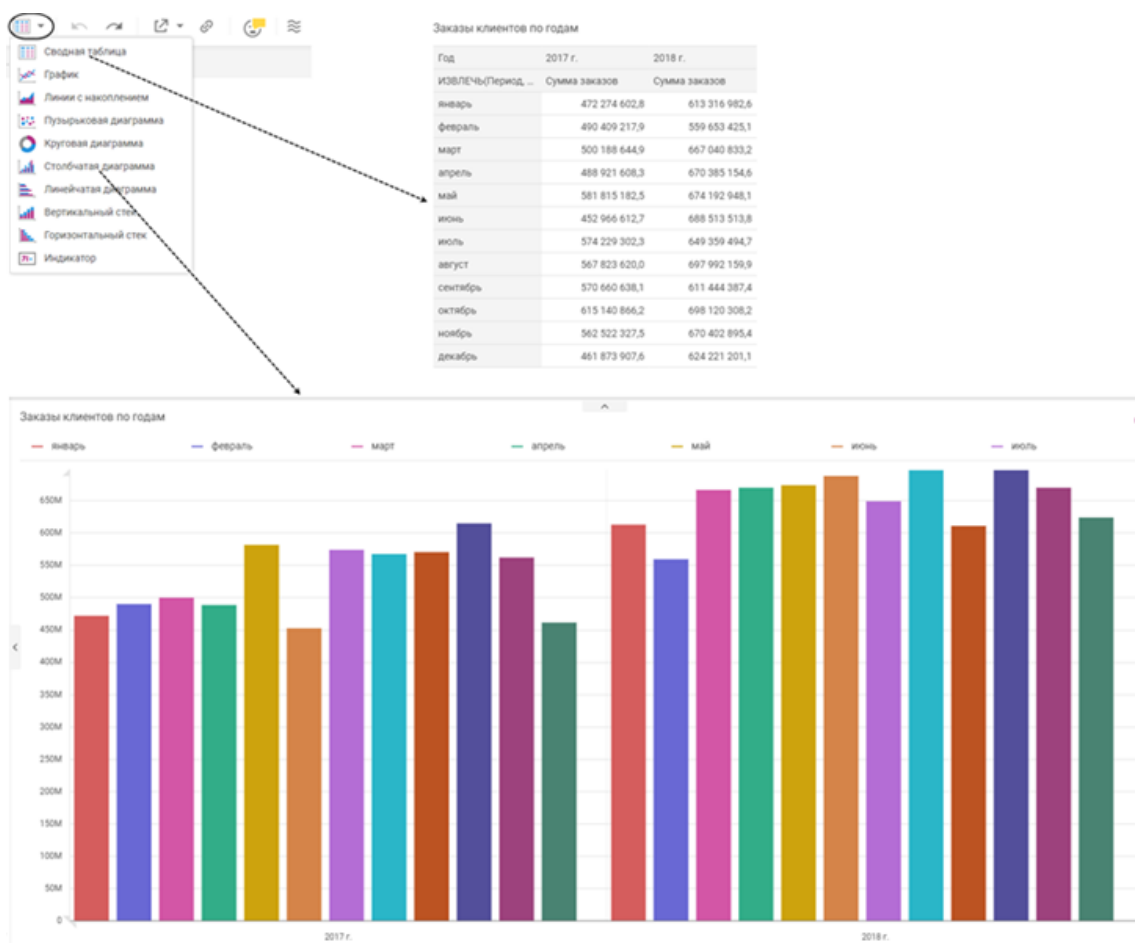


Рис. 9.8. Выбор графического представления диаграммы

Дизайнер диаграммы

После того, как вы выбрали графическое представление диаграммы и установили роли и опции полей в нем, вы можете перейти на закладку Дизайнер диаграммы (со значком малярного валика) на панели источника данных и задать некоторые настройки форматирования диаграммы по своему вкусу.

Переименование и сохранение полей и выражений

Заключительным штрихом разработки диаграмм является переименование и сохранение полей из состава диаграммы. Не стоит пренебрегать этим этапом, так как только после этого ваша работа будет выглядеть законченно и достойно.

Действительно, важно, чтобы другие пользователи «1С:Аналитики», открыв ссылку на вашу диаграмму, смогли без труда разобраться в ней, а также проанализировать данные с разных сторон, используя сохраненные вами фильтры и поля [4].

Надо понимать, что за каждым полем, находящимся на панели состава диаграммы, скрывается выражение, которое это поле представляет. Например, для вывода суммарного количества заказов используется выражение: СУММА(Заказано).

Заголовок поля по умолчанию будет показывать это выражение (полностью или в усеченном виде). Чтобы другие пользователи диаграммы могли сразу понять, о чем речь, лучше менять названия полей на более понятные и скрывать от пользователя детали выражений. А наиболее часто используемые фильтры и поля желательно сохранять для дальнейшего использования.

Собственные поля

При разработке диаграмм часто требуется не только показать информацию, хранящуюся в исходных полях источника данных «1С:Предприятия», но и отобразить собственные поля, полученные в результате вычислений над ними, например показать среднюю цену товаров, прибыль от их продажи и т. п.

В «1С:Аналитике» вы можете создать собственное поле, которое будет вычислять нужное вам выражение [4].

Например, в диаграмме, показывающей плановые и фактические суммы продаж товаров по месяцам, можно рассчитать процент выполнения плана как процентное отношение разницы между фактом и планом продаж за месяц к плановому значению/

Открытие объектов прикладного решения из «1С:Аналитики»

«1С:Аналитика» предоставляет пользователям очень удобную и полезную возможность – просмотр данных, на которых построена диаграмма, непосредственно в том приложении «1С:Предприятия», из которого эти данные получены. Например, вы сформировали диаграмму на основе заказов клиентов и в какой-то момент захотели посмотреть, как выглядит один из этих заказов в информационной базе «1С:Предприятия».

Для этого вы можете в «1С:Аналитике» получить ссылку на документ и перейти по этой ссылке в прикладном решении, или же открыть этот документ непосредственно из «1С:Аналитики».

Работа с дашбордами [4]. Дашборд можно охарактеризовать как некий фотоколлаж, состоящий из набора нескольких виджетов, предназначенный для комплексного представления данных по нужной тематике (рис. 9.9).

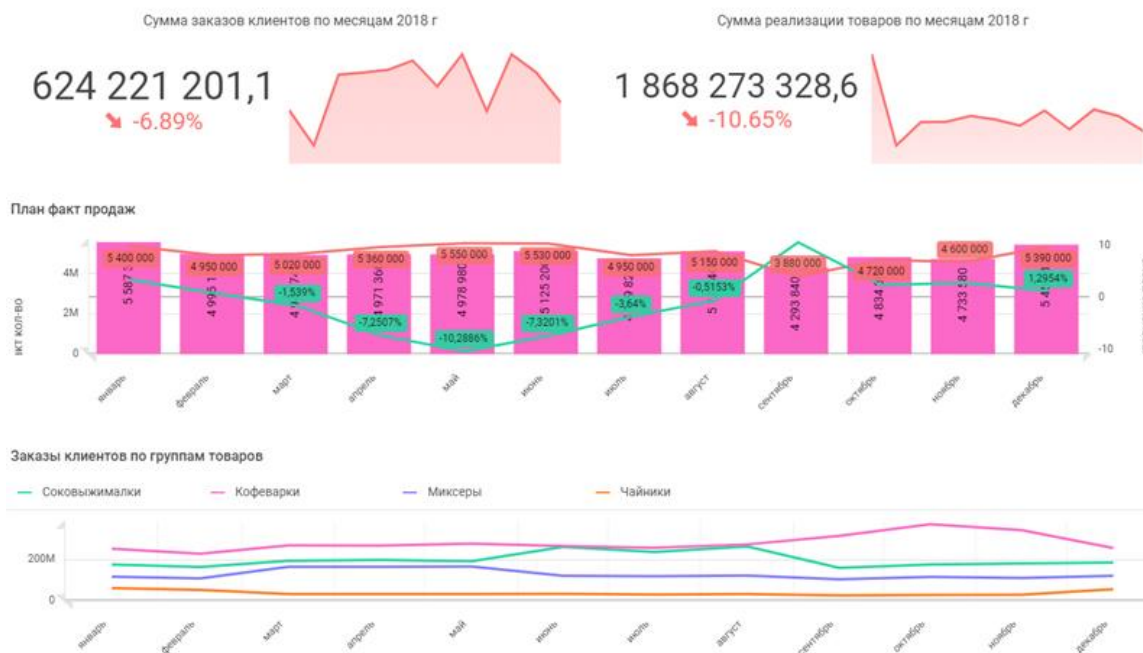


Рис. 9.9. Дашборд с данными о заказах и о реализации товаров за год

Как правило, виджеты содержат отдельные тематически связанные друг с другом диаграммы. Они могут менять свое местоположение и размер [4].

Кроме того, из каждого виджета может быть вызван редактор для соответствующей ему диаграммы. Диаграммы могут иметь общие поля и общие фильтры, которые могут быть применены сразу ко всем диаграммам дашборда.

Для дашборда в целом можно создать интерактивные фильтры, а также раскрыть данные всех диаграмм по какому-либо разрезу (drill down).

Дашборд может содержать такие элементы оформления, как текст, картинки, ссылки на веб-страницы.

Каждый виджет дашборда также может содержать всплывающую подсказку и веб-ссылку, по которой происходит переход при нажатии на виджет.

Условное оформление элементов диаграммы

С помощью языка выражений вы можете оформить по условию различные элементы диаграмм, например, ячейки таблиц, столбцы и сектора диаграмм и т. п. Причем можно выделять столбцы как интерактивно (из рабочей области диаграммы), так и «программно» (по определенному условию), с помощью собственных полей-выражений.

Итак, «1С:Аналитике», это комплексное решение [4], состоящее (рис. 9.10):

- из информационной базы «1С:Предприятия», в которой ведется учет, настраиваются права доступа к данным, права безопасности, хранятся исходные данные, нормативно-справочная информация и т. д.;
- самой «1С:Аналитики», предоставляющей визуальный веб-интерфейс для построения аналитических отчетов на базе «1С:Предприятия»;

– «Дата акселератора», с помощью которого, используя механизм копий баз данных, снимается нагрузка с информационной базы, в которой работают пользователи, и ускоряется выполнение аналитических отчетов.

Информационная база публикуется на веб-сервере, к ней подключается сервер «1С:Аналитики», интегрируется с «1С:Предприятием» и начинает работать с ним как единое целое – получать из него данные, формировать отчеты, получать ссылки на объекты в «1С:Аналитике» и открывать их в «1С:Предприятии». А для ускорения построения аналитических отчетов копия данных разворачивается на «Дата акселераторе», куда перенаправляются все аналитические запросы.

Для развертывания полноценного аналитического решения на базе системы «1С:Аналитика» требуются настройка и установка следующих программных компонентов [4]:

Кластер серверов «1С:Предприятия», в котором размещается конфигурация (прикладное решение) «1С:Предприятия» с данными для анализа внутри «1С:Аналитики».

Веб-сервер для публикации информационной базы «1С:Предприятия» и механизма интеграции с «1С:Аналитикой» (рекомендуется Apache).

Среда исполнения Java (дистрибутив JDK или JRE).

Сервер «1С:Аналитики» (лицензия на «1С:Аналитику» и дистрибутив).

Сервер «Дата акселератора» (для установки «Дата акселератора» серверная лицензия «1С:Предприятия» должна быть уровня КОРП или ПРОФ).

Совместимый веб-браузер у пользователей «1С:Аналитики» (поддерживаются актуальные версии Chrome, Firefox и Safari на момент публикации соответствующей версии «1С:Аналитики»).

В зависимости от наличия ресурсов (оперативной памяти, производительности и т. п.) веб-сервер, сервер «1С:Аналитики» и «Дата акселератор» могут быть развернуты на том же компьютере, что и сервер «1С:Предприятия».

Или же для работы сервера «1С:Аналитики» и «Дата акселератора»

может быть выделен отдельный компьютер. Главное, что эти серверы должны иметь возможность взаимодействия друг с другом по протоколу HTTP.

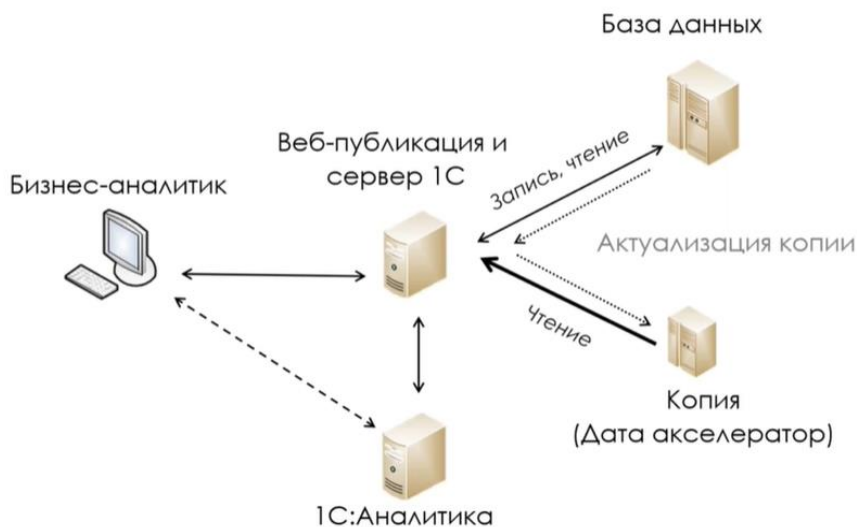


Рис. 9.10. Схема развертывания аналитического решения на базе системы «1С:Аналитика» [4]

Рассмотрим кратко общие приемы работы в «1С:Аналитика» [4]:

1) Просмотр, редактирование, создание и публикация. Просматривать, редактировать и создавать новые диаграммы и дашборды можно на рабочем столе «1С:Аналитики». Если диаграмма или дашборд опубликованы другим пользователем, то они помечаются значком ссылки в списке диаграмм. Нажав на такую диаграмму или дашборд, пользователь может только просматривать ее. Изменить диаграмму или дашборд другого пользователя пользователь не может, но он может сделать копию, изменить ее и опубликовать от своего имени.

Для этого следует открыть диаграмму в редакторе, нажать на значок ссылки справа от ее названия и из выпадающего меню выбрать команду Создать копию.... Затем следует дать диаграмме другое название и нажать Копировать или Копировать и открыть. Команда Копировать добавит копию диаграммы на рабочий стол пользователя, но автором этой диаграммы будете уже он сам и сможет открыть и изменить ее. Команда Копировать и открыть позволит сразу же открыть добавленную диаграмму в режиме редактирования. На это указывает значок карандаша справа от названия диаграммы. После того как закончены изменения диаграммы, следует опубликовать ее, нажав Опубликовать из выпадающего меню под значком карандаша.

После того как диаграмма опубликована, цвет карандаша поменяется с красного на черный. Это значит, что все изменения диаграммы опубликованы. Пользователь может продолжать редактировать диаграмму, при этом цвет карандаша снова поменяется на красный. Это значит, что у диаграммы есть неопубликованные изменения. Эти изменения будут видны только пользователю до тех пор, пока он снова их не опубликует. А текущее (опубликованное) состояние диаграммы пользователь всегда можете увидеть, нажав Просмотр из выпадающего меню под значком карандаша [4].

При нажатии на диаграмму на рабочем столе она открывается всегда в том режиме (просмотра или редактирования), в котором пользователь ее оставил последний раз. Чтобы создать собственную диаграмму, следует нажать кнопку Создать диаграмму в верхнем правом углу рабочего стола, выбрать источник данных, на котором она будет строиться, и т. д.

В процессе редактирования диаграммы запоминаются последние 20 изменений, сделанных до момента опубликования диаграммы. С помощью соответствующих кнопок в командной панели редактора диаграммы можно при необходимости отменять или восстанавливать эти изменения. После того как пользователь закончил изменять диаграмму, следует ее опубликовать, нажав Опубликовать из выпадающего меню под значком карандаша.

Пользователь, с которым автор поделится ссылкой, откроет эту ссылку у себя в браузере и увидит эту диаграмму в режиме просмотра. Затем он может добавить диаграмму другого пользователя на свой рабочий стол, нажав Прикрепить в выпадающем меню под значком ссылки.

Точно так же можно делиться ссылками и на целые папки, содержащие диаграммы. Например, пользователь создал собственную папку для работы в «1С:Аналитике» и наполнили ее диаграммами. Затем он может скопировать

адрес этой папки из адресной строки браузера и послать эту ссылку другому пользователю. Открыв эту ссылку у себя в браузере, другой пользователь может добавить пересланную ему папку со всеми находящимися в ней диаграммами на свой рабочий стол, нажав Прикрепить в выпадающем меню под значком ссылки.

Следует заметить, что обмениваться ссылками на диаграммы можно только в пределах одной информационной базы и в пределах одного сервера «1С:Аналитики». Если различные серверы «1С:Аналитики» подключены к одной и той же информационной базе, то у каждого из них будет свой рабочий стол и они не смогут видеть диаграммы друг друга. Если же один сервер «1С:Аналитики» подключен к разным информационным базам, то для построения диаграммы, полученной по ссылке из одной базы, в другой базе не будет данных, если только эти базы не подобны друг другу.

2) **Настройки. Настройки.** Системное меню «1С:Аналитики», вызываемое по кнопке с тремя волнистыми чертами в правом верхнем углу рабочего стола или редактора диаграммы, включает в себя практически все команды, представленные на командной панели в виде кнопок. Также в системном меню присутствуют следующие функции:

- Опция отправки запросов на подключенный «Дата акселератор». Если опция отключена, то построение данных для диаграмм выполняется на основной базе, подключенной к кластеру «1С:Предприятия». Если опция включена, то «1С:Аналитика» будет запрашивать данные в «Дата акселераторе» – специальной СУБД, расположенной в оперативной памяти. Такие запросы выполняются гораздо быстрее, но требуется предварительная настройка «Дата акселератора» на стороне информационной базы «1С:Предприятия».

- Опция вывода служебной информации. Если опция включена, то внизу диаграммы будет отображаться, на какой базе сформирована диаграмма, сколько времени ушло на ее построение в целом, сколько заняло получение информации из базы данных.




- Опция темной темы. Определяет, в каком стиле будут отображаться служебные панели «1С:Аналитики»: светлое на темном или наоборот.

- Центр оповещений, в котором сохраняются для просмотра последние служебные сообщения.

- Переход в «1С:Предприятие».

- Сведения о программе.

- Выход пользователя из системы.

3) **Поля источника данных.** В «1С:Аналитике» все поля имеют две роли: измерение и факт. В состав диаграммы также входят фильтры. Фильтры определяют, какие данные из информационной базы попадут в диаграмму. Затем все выбранные факты группируются по входящим в состав отчета измерениям. Измерения в «1С:Аналитике» отмечены пурпурным цветом , факты – синеголубым , а фильтры – серо-синим .

Состав полей в источнике соответствует исходным данным объекта из информационной базы системы «1С:Предприятие». Например, если в качестве источника для диаграммы выбрать регистр накопления Заказы клиентов, то

можно увидеть, что «1С:Аналитика» предлагает поля, соответствующие данным регистра (измерениям, ресурсам, реквизитам, стандартным реквизитам) в конфигураторе системы «1С:Предприятие».

Поля на панели источника данных диаграммы отличаются не только цветом своей роли, но и иконкой слева от имени поля, указывающей на тип поля (булево, число, дата).

Если поле имеет ссылочный тип, то иконка приобретает вид галочки вправо, раскрыв которую, вы увидите список полей источника, на который указывает ссылка. Поля этого источника тоже можно использовать в составе диаграммы.

Нажав на три вертикальные точки справа от названия поля, можно вызвать контекстное меню, позволяющее:

- добавить поле в состав диаграммы;
- добавить поле с другой ролью (добавить измерение как факт или факт как измерение);
- показать сразу все элементы иерархического справочника;
- посмотреть описание этого поля в «1С:Аналитике»;
- создать фильтр на основании данного поля;
- создать группировку по полю (если тип поля позволяет это сделать).

Названия полей в панели источника данных соответствуют синонимам реквизитов объекта в конфигурации «1С:Предприятия». Это более понятно и удобочитаемо для пользователей, но при работе с полями в составе диаграммы в редакторе выражений всегда используется имя поля в конфигурации.

Поэтому нужно помнить, что имена полей, используемые в выражениях, должны соответствовать правилам именования полей на встроенном языке «1С:Предприятия» (не содержать пробелов и начинаться с буквы).

Для более быстрого и удобного поиска полей (особенно если источник данных имеет сложную структуру и большое число полей, а также вложенных в них элементов) вверху панели источника данных находится строка поиска.

Справа от названия источника данных диаграммы расположена кнопка со знаком плюс, с помощью которой можно добавить в состав диаграммы собственное поле, описанное в редакторе выражений

4) **Фильтры данных.** Одним из важнейших моментов при создании и редактировании диаграммы является добавление в нее фильтров для отбора данных, которые пользователь хочет получить. Создать фильтр можно из контекстного меню нужного поля на панели источника данных или на панели состава диаграммы, а можно отметить данные для фильтра непосредственно в рабочей области диаграммы.

В окне редактирования фильтра для полей, ссылающихся на справочники, слева вы видите список доступных значений, которые можно добавить в фильтр, а справа – список значений, отобранных в фильтр.

Если справочник иерархический, то доступные для выбора значения справочника будут показаны в виде иерархического дерева, в котором можно

открывать папки для доступа к лежащим в них элементам. При нажатии на элемент или кнопку со знаком «+» элемент будет добавлен в состав фильтра и появится в списке в правой половине окна.

При необходимости пользователь можете отфильтровать данные в диаграмме по значению поля факта. Диалоговое окно создания фильтра стандартно открывается в режиме редактирования диапазона возможных значений. В этом режиме вы можете указать нужные значения показателей, двигая ползунками и ограничивая данные, которые в итоге попадут на диаграмму. Или же можно вручную отредактировать значения в полях от и до.

Кроме того, можно переключить окно редактирования фильтра в режим редактирования выражений, открыв выпадающее меню справа от названия фильтра. В окне выражений пользователь увидит выражение, где в качестве минимального и максимального значения поля факта задан выбранный вами диапазон:

МЕЖДУ(СУММА(Сумма), 22046013703.201, 22046013706.694)

Пользователь может изменить это выражение, например на СУММА(Сумма) > 6000000000.

Такой же эффект можно получить, если в редакторе диапазона указать только нижнюю границу, а справа от верхней границы нажать на знак бесконечности.

При работе с диаграммами часто возникает потребность отфильтровать и посмотреть данные по какому-то фрагменту данных, которые пользователь уже видит на экране. В таких ситуациях часто бывает удобно добавлять фильтры в диаграмму не через диалоговое окно создания фильтра, а непосредственно через саму диаграмму.

Для этого достаточно выделить нужные пользователю данные в рабочей области диаграммы и затем в меню дополнительных действий, появившемся над панелью состава диаграммы, отфильтровать данные, нажав кнопку Оставить или Исключить.

5) Группировка данных. Это удобный, быстрый и эффективный инструмент, предоставляемый системой «1С:Аналитика». Использование группировки позволяет упростить представление данных и их анализ, объединяя в группы похожие элементы и позволяя их анализировать как единое целое. Создать группировку можно из контекстного меню поля (если содержащиеся в нем данные можно сгруппировать) на панели источника данных или же выбирая нужные поля для группировки непосредственно в рабочей области диаграммы.

6) *Графическое представление диаграмм.* Данные диаграммы могут быть представлены в виде: сводной таблицы; графика; линий с накоплением; пузырьковой диаграммы; круговой диаграммы; столбчатой диаграммы; линейчатой диаграммы; вертикального стека; горизонтального стека; индикатора.

При выборе графического представления диаграммы для каждого отображаемого поля нужно указать роль и дополнительные опции на данном виде графика. Состав ролей и опций зависит от вида диаграммы (табл. 9.1). На

панели состава диаграммы слева от названия каждого поля находится иконка, отражающая выбранные пользователем роли и опции поля.

Таблица 9.1

Состав ролей и опций для разных видов диаграмм

Вид диаграммы	Роль и дополнительные опции
1	2
Сводная таблица	<p>В табличном представлении данных в диаграмме у измерений и фактов может быть лишь одна роль – отображение в строках или в колонках. Для всех фактов эта роль должна быть одинаковой, так как нельзя часть фактов показывать строками, а часть колонками. Поэтому при изменении вида отображения у одного поля факта у других фактов отображение меняется автоматически.</p> <p>Например, данные заказов клиентов можно представить в виде сводной таблицы, в которой годы и поля фактов (количество и сумма заказов) показаны в колонках, а месяцы – в строках таблицы. Или же можно вывести данные «с точностью до наоборот», то есть месяцы показать в колонках, а годы и поля фактов – в строках таблицы.</p>
Столбчатая диаграмма	<p>Если данные представляются в виде столбчатой диаграммы, то для измерений появляются следующие роли и опции:</p> <p>Ось X. Измерение отображается на оси X – например, по горизонтали идет шкала месяцев, по которым строится диаграмма.</p> <p>Группы. Измерение отображается отдельными столбцами для каждого значения – например, для каждого года значение показывается отдельным столбцом.</p> <p>Цвет. Для измерения используется одинаковый цвет для серии столбцов, или для каждого столбца генерируется свой собственный цвет.</p> <p>Маркеры. На столбце измерения маркером отображается значение – например, год, для которого указано значение.</p> <p>Для фактов появляются следующие роли и опции:</p> <p>Левая ось Y. Факт отображается столбцом на левой шкале диаграммы.</p> <p>Линии (левая ось Y). Факт отображается линией на левой шкале диаграммы.</p> <p>Линии (правая ось Y). Факт отображается линией на правой шкале диаграммы.</p> <p>Маркеры, Маркеры (линии). Значение факта дублируется на диаграмме текстом.</p> <p>Роли измерения Ось X и Группы являются взаимозависимыми. Поэтому, если одному измерению вы присвоили роль Ось X, у</p>

Продолжение табл. 9.1.

1	2
	другого измерения роль автоматически меняется на Группы, и наоборот.
Круговая диаграмма	Если данные представляются в виде круговой диаграммы, то для измерений появляются две роли: Пироги и Сектора. При этом, если одному измерению вы присвоили роль Пироги, у другого измерения роль автоматически меняется на Сектора, и наоборот.
График	<p>Если данные в диаграмме представляются в виде графика, то для измерений появляются следующие роли:</p> <p>Ось X. Измерение отображается на оси X – например, по горизонтали идет шкала месяцев, по которым строится график.</p> <p>Линии. Измерение отображается отдельными линиями для каждого значения – например, для каждого года значение показывается отдельной линией.</p> <p>Роли измерений также изменяются автоматически. То есть, если одному измерению вы присвоили роль Ось X, у другого измерения роль автоматически меняется на Линии, и наоборот.</p> <p>Для фактов появляются следующие роли и опции:</p> <p>Левая ось Y. Факт отображается линией на левой шкале графика.</p> <p>Правая ось Y. Факт отображается линией на правой шкале графика.</p> <p>Интерполяция. Значения факта выводятся в виде сглаженной кривой.</p> <p>Маркеры. Значение факта дублируется на графике диаграммы текстом.</p>
Индикатор	<p>Диаграмма в виде индикатора используется в первую очередь в составе дашборда, когда среди прочих элементов дашборда надо показать или выделить несколько конкретных показателей.</p> <p>Диаграмму-индикатор можно создать, непосредственно выбрав этот вид диаграммы из выпадающего меню на командной панели. Для более наглядного представления индикатора можно на закладке Дизайнер диаграммы (со значком малярного валика) включить опцию Спарклайн и увидеть небольшой график изменения показателя.</p> <p>Таким образом, посмотрев на это компактное представление, можно быстро понять, что происходит с показателем и требует ли этот показатель более внимательного изучения.</p> <p>Другой вариант создания диаграммы-индикатора – непосредственно из данных диаграммы. Для этого нужно выбрать поле, по которому вы хотите вывести показатель, и в меню дополнительных действий, появившемся над панелью состава диаграммы, нажать кнопку Индикатор.</p>

Продолжение табл. 9.1.

1	2
	<p>В результате в состав диаграммы будут добавлены нужные фильтры и на индикаторе будет показано только выбранное вами значение.</p> <p>В составе дашборда такие диаграммы-индикаторы могут упрощать восприятие нужной информации и комбинироваться с диаграммами с детальной информацией.</p>

7) **Переименование и сохранение полей и выражений.** Переименование и сохранение полей выполняются в редакторе поля, который вызывается с помощью команды Редактировать в контекстном меню поля на панели состава диаграммы. В редакторе поля справа от его названия находится выпадающее меню, с помощью которого пользователь может переименовать и сохранить поле.

8) **Работа с выражениями.** В «1С:Аналитике» пользователь может создать собственное поле, которое будет вычислять нужное вам выражение. «1С:Аналитика» поддерживает довольно много различных функций для конструирования выражений: функции агрегирования, функции для работы с датами, оконные функции и т. д.

9) **Работа со ссылками.** Каждый объект базы данных (элемент справочника, документ, счет и т. д.) в «1С:Предприятии» характеризуется уникальной совокупностью данных и имеет свой внутренний идентификатор. Различные объекты базы данных всегда будут иметь разные внутренние идентификаторы. Этот уникальный идентификатор можно получить и использовать в качестве ссылки на объект в «1С:Аналитике».

Во-первых, ссылка на объект может быть использована для открытия объектов прикладного решения из «1С:Аналитики». Во-вторых, в языке выражений можно сравнивать ссылочные объекты, просто указав в кавычках наименование объекта.

10) **Детализация данных и открытие объектов прикладного решения из «1С:Аналитики».** Например, пользователь сформировал диаграмму на основе заказов клиентов и в какой-то момент захотел детально посмотреть, из каких конкретно документов Заказ клиента состоят полученные вами данные. Для этого в меню дополнительных действий, появившемся над панелью состава диаграммы, нужно нажать кнопку Раскрыть и из выпадающего списка измерений выбрать Заказ клиента. В результате нужное измерение и фильтр будут добавлены в состав диаграммы и данные будут детализированы до конкретных документов. После этого можно выделить нужный документ и в меню дополнительных действий нажать Скопировать ссылку

Итак, «1С:Аналитика» предназначена для решения следующих задач:

- Быстрое получение и анализ информации в базе «1С:Предприятия».
- Возможность использовать штатные структуры метаданных «1С:Предприятия», названия объектов из используемой конфигурации.
- Возможность смотреть отчеты через браузер на компьютере или на мобильном устройстве.
- Возможность идти от общих аналитических данных до конкретного документа в информационной базе «1С:Предприятия».
- Возможность интерактивной работы с данными (формирование отчетов «на лету»).

Это мощный и гибкий механизм, который позволяет:

- Создавать с нуля аналитические отчеты на основе данных внутри платформы «1С:Предприятие».
- Работать с уже созданными диаграммами и дашбордами и делиться ссылками на них с другими пользователями.
- «На лету» изменять состав отчетов и фильтры, по которым отбираются данные.

Специалисты отмечают, что «1С:Аналитика» – не замена, а дополнение уже существующей в «1С:Предприятии» системы компоновки данных. Оба этих компонента работают с одними и теми же данными.

Система компоновки данных больше ориентирована на разработчика. Разработчик создает отчет, пользователь запускает его и может время от времени изменять настройки этого отчета при необходимости.

«1С:Аналитика», напротив, ориентирована на то, что пользователь сам, простым и понятным образом, создает нужные ему отчеты. В «1С:Аналитике» можно быстро и легко «покрутить» данные с разных сторон, в виде различных диаграмм и сводных аналитических таблиц. Кроме того, «1С:Аналитика» интегрирована с «Дата акселератором», за счет чего снижается нагрузка на информационную базу и ускоряется выполнение аналитических запросов для построения отчетов.

«1С:Аналитика» прежде всего востребована:

- Сотрудниками, отвечающими за анализ данных в компании.
- Руководителями, которые хотят понимать, что реально происходит в компании, по данным прикладного решения.
- Пользователями, которым необходима постоянная интерактивная работа с данными и возможность гибкого построения разнообразных отчетов.

Заключение

Материал лекционных занятий предусматривает получение обучающимися теоретических представлений о методах, механизмах и технологиях реализации систем бизнес-аналитики и модулей анализа данных в приложениях на платформе «1С:Предприятие», использование которой для разработки российского программного обеспечения соответствует политике импортозамещения, изучение основных принципов разработки функционала бизнес-аналитики в корпоративных информационных системах на платформе «1С:Предприятие».

В результате освоения лекционного материала:

- сформированы знания возможностей разработки на базе отечественной технологической платформы "1С:Предприятие" систем бизнес-аналитики и модулей интеллектуального анализа больших данных;

- получены углубленные знания механизма анализа данных и прогнозирования в «1С:Предприятие» для реализации в корпоративных информационных системах инструментов выявления закономерностей в больших объемах информации;

- получены знания о механизмах интеграции корпоративных информационных системы на базе платформы «1С:Предприятие» с другими информационными системами, форматы файлов обмена

- получены знания, как применять методы и инструменты анализа данных в решении задач профессиональной деятельности с использованием систем бизнес-аналитики на платформе «1С:Предприятие»;

- получены знания основ применения языка запросов и системы компоновки данных для разработки сложных аналитических отчетов;

- сформированы профессиональные компетенции по использованию механизма запросов и системы компоновки данных для реализации сложных аналитических отчетов;

- сформированы профессиональные компетенции управления разработкой архитектуры комплексных систем бизнес-аналитики со стороны заказчика.

Библиографический список

1. <https://its.1c.ru> – сайт Информационно-технологического сопровождение пользователей системы «1С:Предприятие».
2. Профессиональная разработка в системе «1С:Предприятие»: в 2 т. / 2-е изд.– М.: 1С-Паблишинг, 2012.–Т.1.–690с.
3. Профессиональная разработка в системе «1С:Предприятие»: в 2 т. / 2-е изд.– М.: 1С-Паблишинг, 2012.–Т.2.–683с.
4. Хрусталева Е.Ю. ВІ-система в «1С:Предприятие 8».– М.: 1С-Паблишинг.– 2021.– 208с.
5. Хрусталева Е.Ю. Разработка сложных отчетов в «1С:Предприятие 8». Система компоновки данных.– М.: 1С-Паблишинг.– 2016.– 458с.
6. Хрусталева Е. Ю. Язык запросов «1С:Предприятия 8».– М.: 1С-Паблишинг, 2013.– 369с.