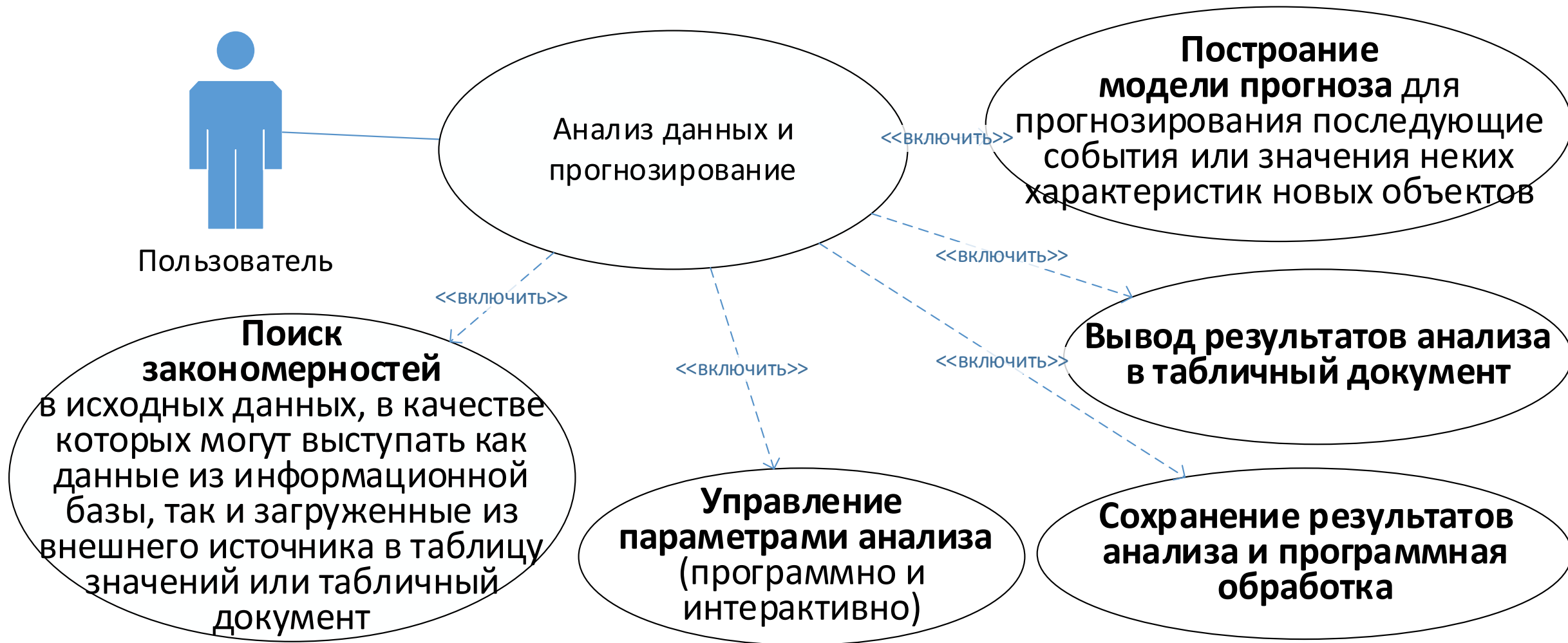


- ❑ *Общая схема работы механизма анализа и прогнозирования данных.*
- ❑ *Основные объекты механизма.*
- ❑ *Типы анализа данных:*
 - *общая статистика,*
 - *поиск ассоциаций,*
 - *поиск последовательностей,*
 - *кластерный анализ,*
 - *дерево решений.*
- ❑ *Модели прогноза.*

Возможности механизма анализа данных и прогнозирования



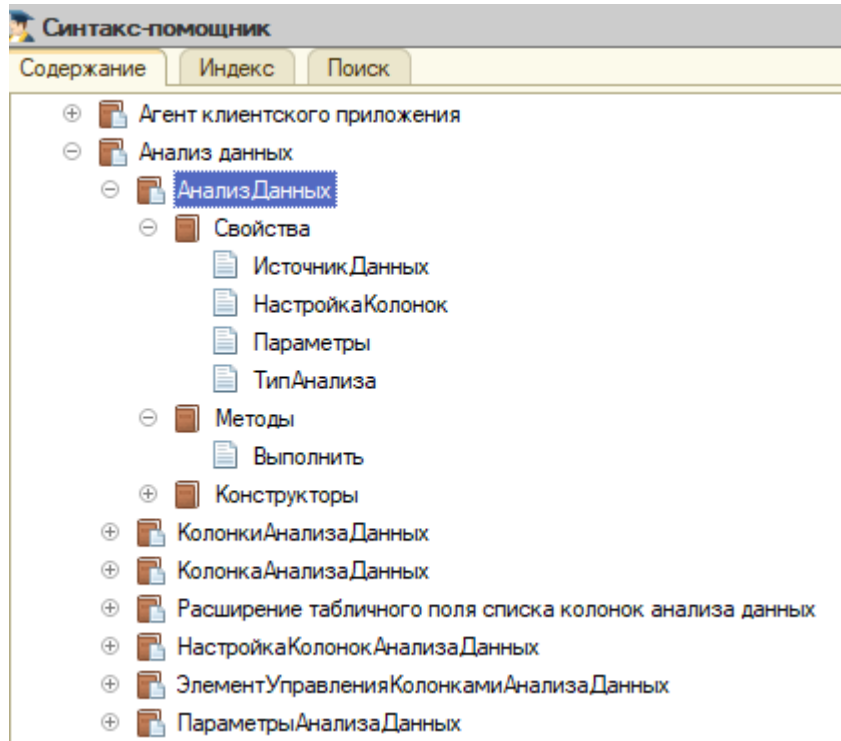
Пользователь



Механизм анализа и прогнозирования данных в «1С:Предприятие»



Общий объект АнализДанных



Общий объект АнализДанных предназначен для выполнения анализа данных (кластерный анализ, поиск ассоциативных правил, поиск последовательностей и другие виды анализа). Результат работы зависит от типа производимого анализа.

Конструктор объекта:

Анализ = Новый АнализДанных;

Свойства объекта:

ИсточникДанных – Содержит источник данных анализа, например. ТаблицаЗначений; РезультатЗапроса; ОбластьЯчеекТабличногоДокумента
НастройкаКолонок – Содержит настройку колонок источника данных. Это 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 – Параметры анализа данных. Состав параметров зависит от типа анализа.

ТипАнализа – Тип производимого анализа. Одно из:

Тип("АнализДанныхКластеризация"),

Тип("АнализДанныхПоискАссоциаций"),

Тип("АнализДанныхПоискПоследовательностей"),

Тип("АнализДанныхДеревоРешений"),

Тип("АнализДанныхОбщаяСтатистика"),

Методы:

Выполнить() – Выполняет анализ и получает результат. Возвращаемое значение результата зависит от свойства

ТипАнализа

Типы анализа данных механизма анализа данных и прогнозирования платформы «1С:Предприятие»

- Общая статистика.
- Поиск ассоциаций.
- Поиск последовательностей.
- Кластерный анализ.
- Дерево решений.

Общая статистика

Представляет собой механизм для сбора информации о данных, находящихся в исследуемой выборке. Этот тип анализа предназначен для предварительного исследования анализируемого источника данных. Анализ показывает ряд характеристик непрерывных и дискретных полей. Непрерывные поля содержат такие типы как Число, Дата. Для остальных типов используются дискретные поля. При выводе отчета в табличный документ заполняются круговые диаграммы для отображения состава полей.

Информация о данных

Количество объектов: 20

Непрерывные поля

Поле	Значений	Минимум	Максимум	Среднее	Размах	Стд. откл.
Количество	20	1	4	1,6000	3	0,820
Сумма	20	2 500	28 400	11 757,5000	25 900	8 294,279

Дискретные поля

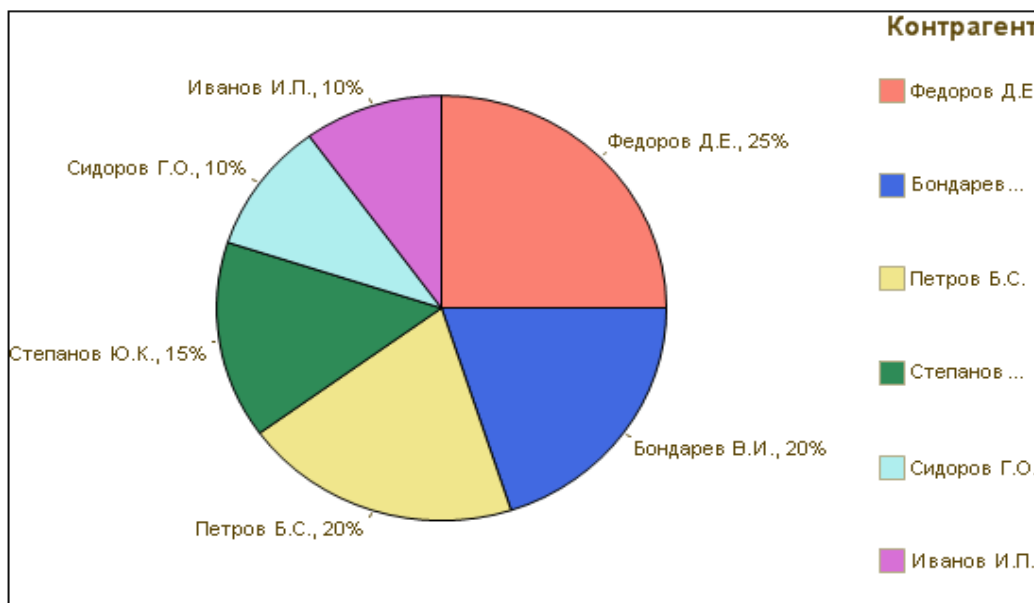
Контрагент

Количество значений: 20
Количество уникальных значений: 6
Мода: Федоров Д.Е.

Таблица частот

Значение	Частота	Относительная частота	Накопленная частота	Накопленная относительная частота
Федоров Д.Е.	5	25,00	5	25,00
Бондарев В.И.	4	20,00	9	45,00
Петров Б.С.	4	20,00	13	65,00
Степанов Ю.К.	3	15,00	16	80,00
Сидоров Г.О.	2	10,00	18	90,00
Иванов И.П.	2	10,00	20	100,00

Диаграмма частот



Типы переменных (данных) в статистике

- ❑ *Номинальные данные* – данные в виде имени, ярлыка или условного кода. Пример, пол, мужской или женский.
- ❑ *Порядковые данные* – порядок данных, в котором большее значение, соответствует большему проявлению какого-либо признака. Могут быть ранжированы. Но интервал между значениями не может быть выражен количественно. Пример, степень ожога.
- ❑ *Интервальные данные* – данные расположены в осмысленном порядке, имеющие равный интервал между измерениями. Пример, температурная шкала.
- ❑ *Непрерывные данные* – могут принимать любое значение или в определенном диапазоне. Пример, рост, ожидается что он будет в диапазоне.
- ❑ *Дискретные данные* – определенные значения, между которыми существуют четкие границы. Пример, количество детей, 1, 2, 3 ребенка, но не 3,5.

Меры центральной тенденции

- ❑ Частота – как часто встречается (повторяется) значение в данных.
- ❑ Среднее – сумма всех значений деленное на их количество.
- ❑ Мода – чаще всего встречающиеся значения.
- ❑ Медиана – значение которое, делит упорядоченные множество пополам (значение из середины данных). Если значений четное количество, тогда используется среднее от двух ближайших значений

Меры изменчивости данных

- **Размах** – разница между самым большим и самым маленьким значением.
- **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** – разность конкретного значения от среднего.
- **Дисперсия** – Сумма квадратов отклонений, деленная на их количество.
- **Стандартное отклонение** – квадратный корень из дисперсии.

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. 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.

Модель прогноза (поиск ассоциаций): Ассоциации товарных подгрупп в сделках с юр. лицами

Наименование: Ассоциации товарных подгрупп в сделках с юр. лицами Код: 48

Тип анализа: Поиск ассоциаций Количество анализируемых факторов: 2

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

OK Записать Закрыть

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 искомых последовательностей, а также регулировать точность получаемых результатов.

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	

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 из 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 ним

Модель прогноза (кластеризация): Сегментация клиентов

Действия: [Настройка анализа] [Выполнить анализ] [Настройка прогноза] [Выполнить прогноз]

Наименование: Сегментация клиентов Код: 31

Тип анализа: Кластеризация Количество анализируемых факторов: 22

Центры кластеров | Настройки колонок аналитической выборки | Описание

Поле	Значение	Кластер_1, доля: 51,7%	Кластер_2, доля: 20,7%	Кластер_3, доля: 17,2%	Кластер_4, доля: 6,9%	Кластер_5, доля: 1,8%
ABC класс покупателя						
ABC класс покупателя	A - класс		50			
ABC класс покупателя	B - класс		50			
ABC класс покупателя	C - класс	100		100	100	
Выручка к событиям		18 489,31	94 940,01	27 420,17	4 585 453,87	
Группа видов деятельно...						
Группа источников						
Класс продаж		0,02	20,17	10,2	15	

Межкластерные расстояния | Значения прогнозных колонок | Факторы, определившие кластеризацию

Фактор	Рейтинг	Кластер	Доля, % % оши...	Комбинация условий
Продаж в месяц	37	1	51,7	
Сумма оплаты	28	Решение 1	51,7	ОплатаКВыручке > 0,92, ПродажВМесяц > 0,07, СуммаОплаты =...
Оплата к выручке	20	Решение 1		ОплатаКВыручке > 0,92
		Решение 1		ПродажВМесяц > 0,07
		Решение 1		СуммаОплаты =< 3 607 090,47
		2	20,7	

OK Записать Закрыть

Дерево решений

Тип анализа
дерево решений
позволяет
построить
иерархическую
структуру
классифицирующ
их правил,
представленную в
виде дерева

Модель прогноза (дерево решений): Срок выполнения заказа поставщиком

Действия ▾ [Иконки] ? Настройка анализа Выполнить анализ Настройка прогноза Выполнить прогноз

Наименование: Код:

Тип анализа: Количество анализируемых факторов:

Закономерности

Рейтинг факторов:

Фактор	Рейтинг
Задержка поставки план	48
Колво позиций	40
День недели даты поступления	30
Закупки номенклатура сумма	12
Контрагент	8

Дерево "Решение-Комбинация условий":

Решение	Доля, %	% ошибки	Комбинация условий
[-] 2	47,9		
[-] Решение 2	37,5		Колво_Позиций =< 8,5, ДеньНеделиДатыПоступлен...
[-] Решение 2			Колво_Позиций =< 8,5
[-] Решение 2			ДеньНеделиДатыПоступления > 3
[-] Решение 2			ЗадержкаПоставкиПлан =< 3
[+] Решение 1	10,4		Колво_Позиций =< 8,5, ДеньНеделиДатыПоступлен...
[+] 3	20,8		
[+] 0	14,6		
[+] 10	12,5		
[+] 7	4,2		

OK Записать Закреть

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

Прогноз: Риск закрытия заказа клиентом

Прогнозирование - процедура оценки значений прогнозных колонок источника данных для прогноза на основе закономерностей, выявленных и сохраненных в модели прогноза.

Прогноз

Контрагент	Ответственный	ЗАКАЗ	Причина закрытия...	Причина закрытия вероятнос...	Упущенный доход
000 "Универсам...	Иванов Петр Федоро...	Заказ покупателя ТК000176 ...	Отказ контрагента	100	190 000
000 "Универсам...	Иванов Петр Федоро...	Заказ покупателя ТК000174 ...	Отказ контрагента	100	150 000
000 "Универсам...	Иванов Петр Федоро...	Заказ покупателя ТК000111 ...	Отказ контрагента	100	40 120
000 "Универсам...	Иванов Петр Федоро...	Заказ покупателя ТК000125 ...	Отказ контрагента	100	49 500
000 "Универсам...	Иванов Петр Федоро...	Заказ покупателя ТК000126 ...	Отказ контрагента	100	10 000
000 "Универсам...	Иванов Петр Федоро...	Заказ покупателя ТК000129 ...	Отказ контрагента	100	190 000
000 "Супермарк...	Иванов Петр Федоро...	Заказ покупателя ТК000161 ...	Просроченные за...	87,5	74 375
000 "Супермарк...	Иванов Петр Федоро...	Заказ покупателя ТК000123 ...	Просроченные за...	87,5	148 750
000 "Супермарк...	Иванов Петр Федоро...	Заказ покупателя ТК000124 ...	Просроченные за...	87,5	74 375
000 "Супермарк...	Иванов Петр Федоро...	Заказ покупателя ТК000151 ...	Просроченные за...	87,5	236 250
Колво разл = 5		Колво = 14	Колво разл = 2	Мин = 62,5	Сум = 1 236 994,05

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.

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.



- *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* — «Деревья 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.

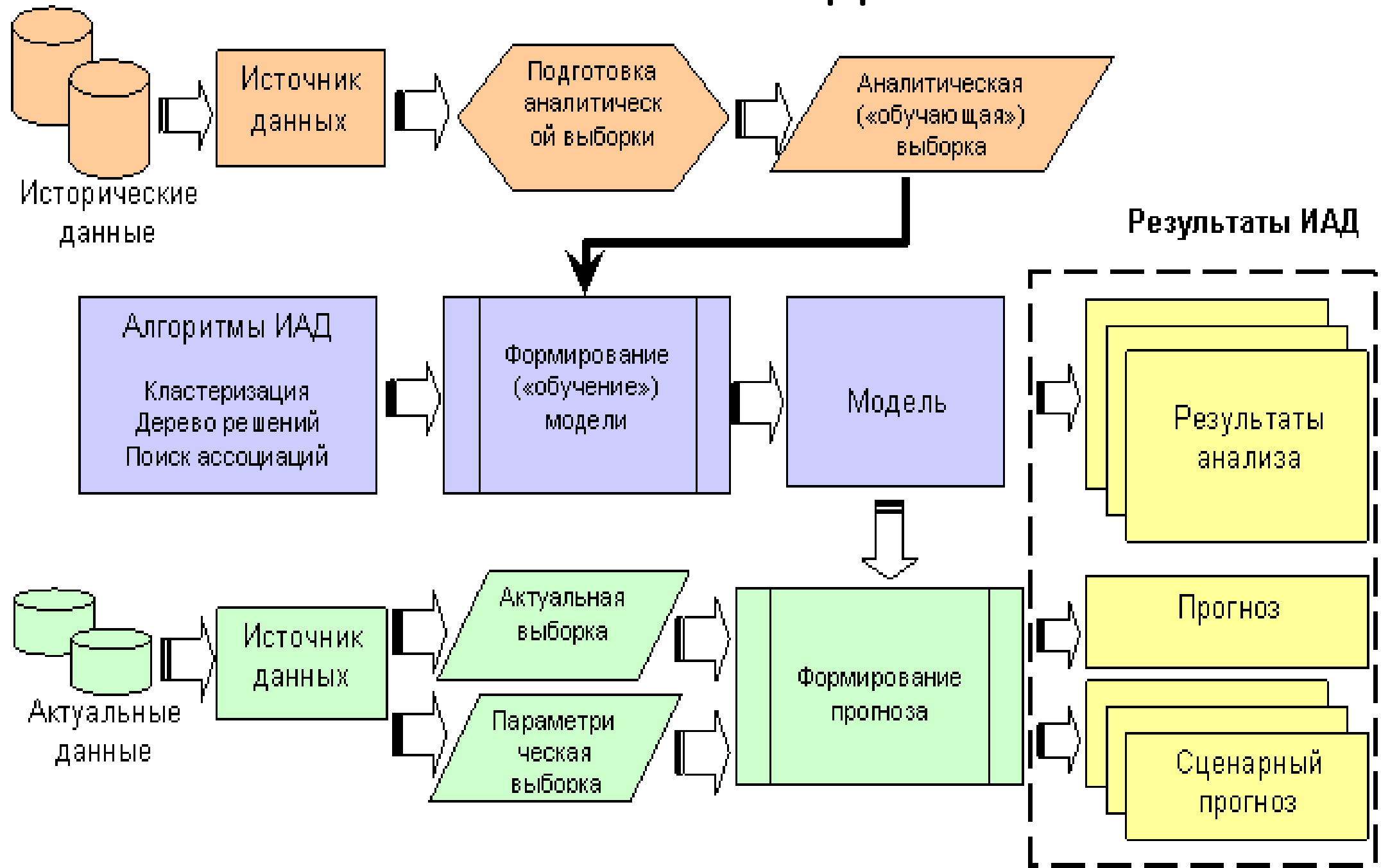
Итак, Механизм анализа данных и прогнозирования – это один из механизмов формирования экономической и аналитической отчетности. Он предоставляет пользователям (экономистам, аналитикам и т.д.) возможность осуществлять поиск неочевидных закономерностей в данных, накопленных в информационной базе.

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 и прогнозирования — это набор взаимодействующих друг с другом объектов встроенного языка, дающих возможность разработчику использовать его составные части в произвольной комбинации в любом прикладном решении. Встроенные объекты позволяют легко организовать интерактивную настройку параметров анализа пользователем, а также выводить результат анализа в удобной для отображения форме в табличный документ. Важно и то, что механизм может работать с данными, полученными как из информационной базы «1С», так и из внешних источников (в последнем случае — предварительно загруженными в таблицу значений или табличный документ)

Механизм анализа данных

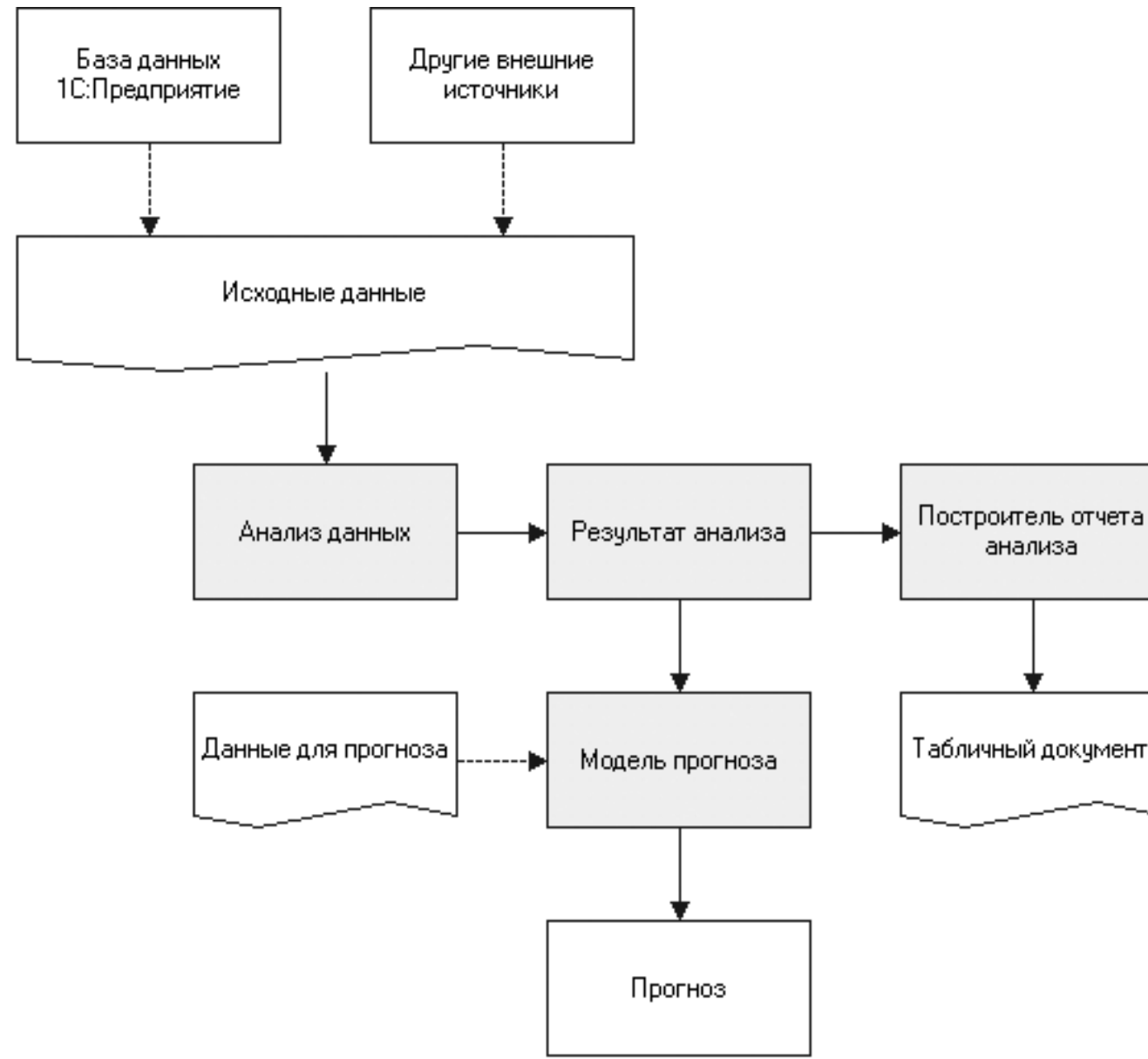


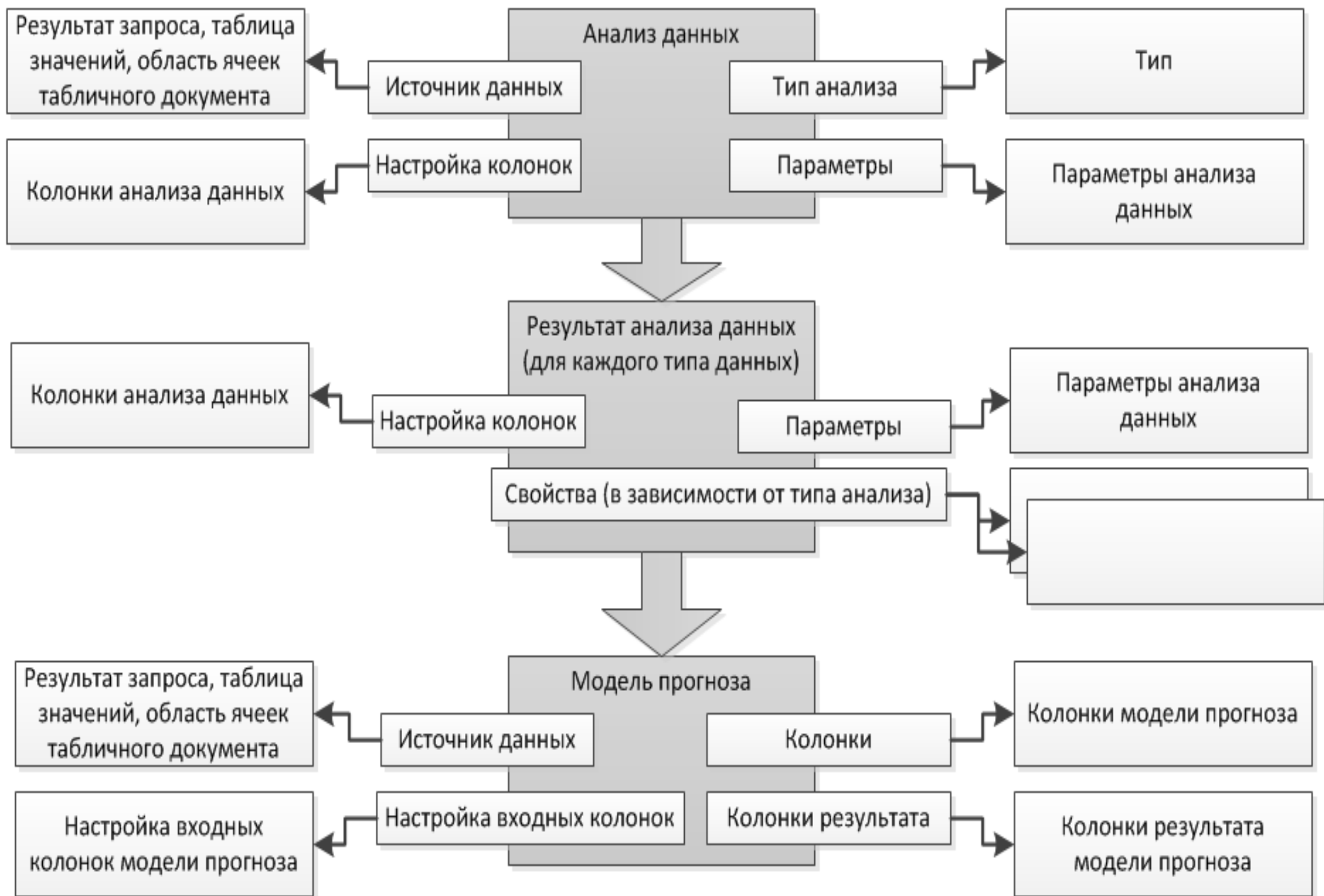
Тема 2. Алгоритмы машинного обучения. Типы анализа «Поиск ассоциаций» и «Поиск последовательностей»



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

Общая схема работы механизма анализа и прогнозирования данных





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

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.

Тип анализа «Поиск ассоциаций»

Данный тип анализа осуществляет поиск часто встречаемых вместе комбинаций объектов или значений характеристик. С его помощью можно определять группы одновременно покупаемых товаров, выявлять наиболее привлекательные источники информации (в процессе «оптимизации» затрат на них) и т. п.



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)

Схема процесса проведения анализа Поиск ассоциаций



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.

Пример кода для анализа Поиск ассоциаций

```
&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
```

Результат анализа Поиск ассоциаций

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

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

Группа товаров

Ассоциативные правила

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			

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

Схема анализа «Поиск последовательностей»



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 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.

Пример кода анализа «Поиск последовательностей»

```
&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
```

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 искомых последовательностей.

■ *ПоискПоИерархии* – признак осуществления поиска по иерархии (распространяется на колонки с типом Элемент).

Значения системного перечисления ТипЕдиницыИнтервалаВремениАнализаДанных

Значения системного перечисления
«ТипЕдиницыИнтервалаВремениАнализаДанных»

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

Информация о данных

Количество элементов: 12

Количество последовательностей: 6

Результат анализа

Найдено последовательностей: 2

Результат анализа

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.

Найденные последовательности

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).

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



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

Схема механизма проведения кластерного анализа



Код проведения кластерного анализа

```
&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
```

Начальные установки анализа данных

Параметры анализа

Количество искомых кластеров:	3
Стандартизация:	Стандартизировать
Мера расстояния:	Евклидова метрика в квадрате
Метод кластеризации:	Метод центра тяжести

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

Исходная выборка данных

Контрагент	Количество розничных точек	Количество автомобилей	Время работы организации	Время заключения договора	Вид договора	Состояние взаимоотношений
ЗАО Игорь	1	0	Меньше года	Меньше года	Дилер	Несоблюдение договора
ЗАО ТогрМебель	15	4	От трех до десяти лет	Меньше года	Дистрибьютор	Прекращение контрагентом
ЗАО ТогрМебель	1	10	От трех до десяти лет	От года до трех	Дистрибьютор	Прекращение контрагентом
ИЧП Дубрава	1	1	От года до трех	Меньше года	Дилер	Прекращение контрагентом
Магазин № 15	1	1	Свыше десяти лет	От трех до десяти лет	Постоянный партнер	Не прекращены
ООО Гросс	3	2	Меньше года	Меньше года	Постоянный партнер	Не прекращены
ООО Интарис	7	3	От трех до десяти лет	От года до трех	Постоянный партнер	Прекращение контрагентом
ООО ТогрТрест	2	2	Свыше десяти лет	От трех до десяти лет	Постоянный партнер	Не прекращены
ПБОЮЛ Курочкин	0	1	Меньше года	Меньше года	Дилер	Не прекращены

Результат анализа

Информация о данных

Количество объектов: 9

Результат анализа

Найдено кластеров: 3

Кластеры

№ кластера	Количество объектов	Процент
1	6	66,67
2	2	22,22
3	1	11,11

Результат анализа

Центры кластеров

Поле \ Кластер	1	2	3
Ссылка = ЗАО ТоргМебель			1,0000
Ссылка = ЗАО МебельКрафт		0,5000	
Ссылка = ООО Интарис		0,5000	
Ссылка = ПБОЮЛ Курочкин	0,1667		
Ссылка = ООО ТоргТрест	0,1667		
Ссылка = Магазин № 15	0,1667		
Ссылка = ЗАО Игорьь	0,1667		
Ссылка = ООО Гросс	0,1667		
Ссылка = ИЧП Дубрава	0,1667		
КоличествоРозничныхТочек	1,3333	11,0000	1,0000
КоличествоАвтомобилей	1,1667	3,5000	10,0000
ВремяРаботыОрганизации = От года до трех	0,1667		
ВремяРаботыОрганизации = Меньше года	0,5000		
ВремяРаботыОрганизации = Свыше десяти лет	0,3333		
ВремяРаботыОрганизации = От трех до десяти лет		1,0000	1,0000
ВремяЗаключенияДоговора = От трех до десяти лет	0,3333		
ВремяЗаключенияДоговора = Меньше года	0,6667	0,5000	
ВремяЗаключенияДоговора = От года до трех		0,5000	1,0000
ВидДоговора = Постоянный партнер	0,5000	0,5000	
ВидДоговора = Дилер	0,5000		
ВидДоговора = Дистрибьютор		0,5000	1,0000
ПрекращениеОтношений = Не прекращены	0,6667		
ПрекращениеОтношений = Несоблюдение договора	0,1667		
ПрекращениеОтношений = Прекращение контрагентом	0,1667	1,0000	1,0000

Расстояния между кластерами

Кластеры	1	2	3
1		8,7740	15,4524
2	8,7740		11,5370
3	15,4524	11,5370	

Используемые метрики

Евклидова метрика

В данной метрике расстояние между двумя объектами вычисляется по формуле:

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

где:

- 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.

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:

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

Евклидова метрика в квадрате

В данной метрике расстояние между двумя объектами вычисляется по формуле:

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

где:

- 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.

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:

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

Метрика города

В данной метрике расстояние между двумя объектами вычисляется по формуле:

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

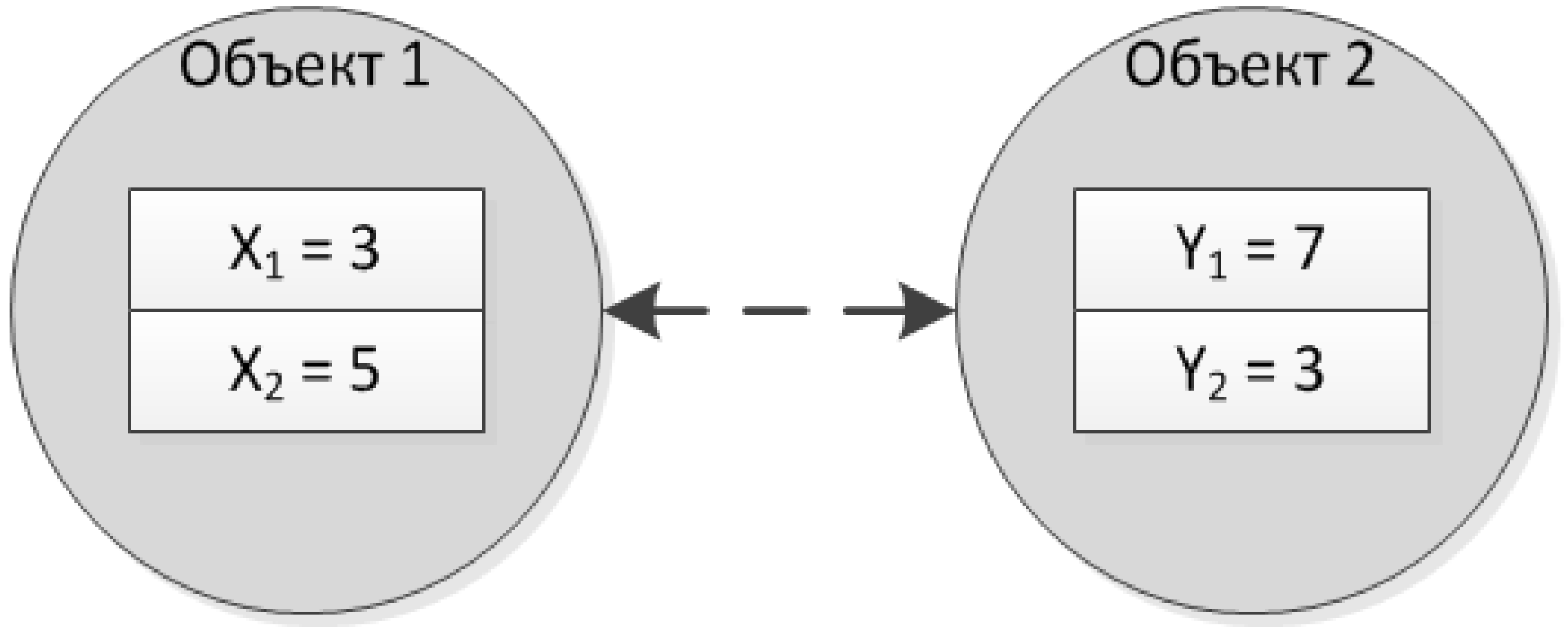
где:

- 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.

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

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

Характеристики объектов



Метрика доминирования

В данной метрике расстояние между двумя объектами вычисляется по формуле:

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

где:

- 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.

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

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

Методы кластеризации

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

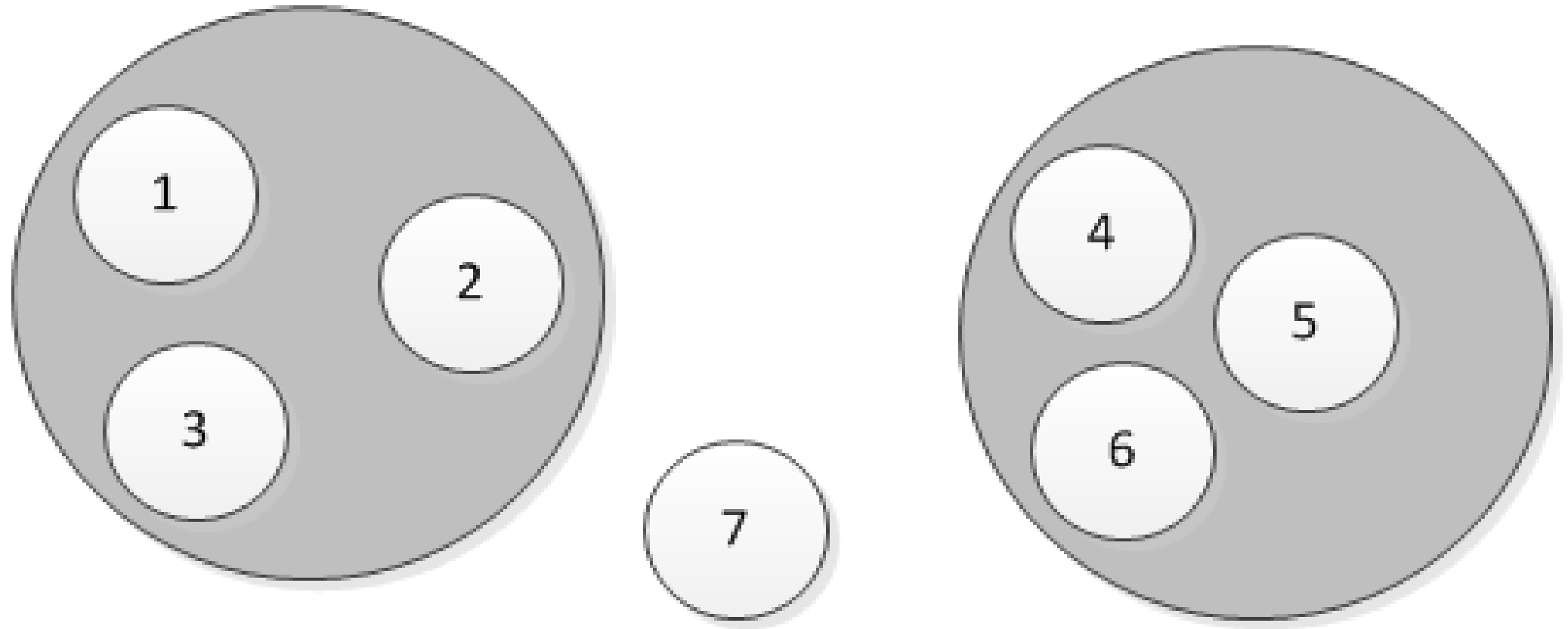
Можно сказать, что целью любого алгоритма кластеризации является:

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

Различия между методами будем рассматривать на объектах, представленных на рисунке на следующем слайде.

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

Группы объектов



Добавляемый
объект

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.

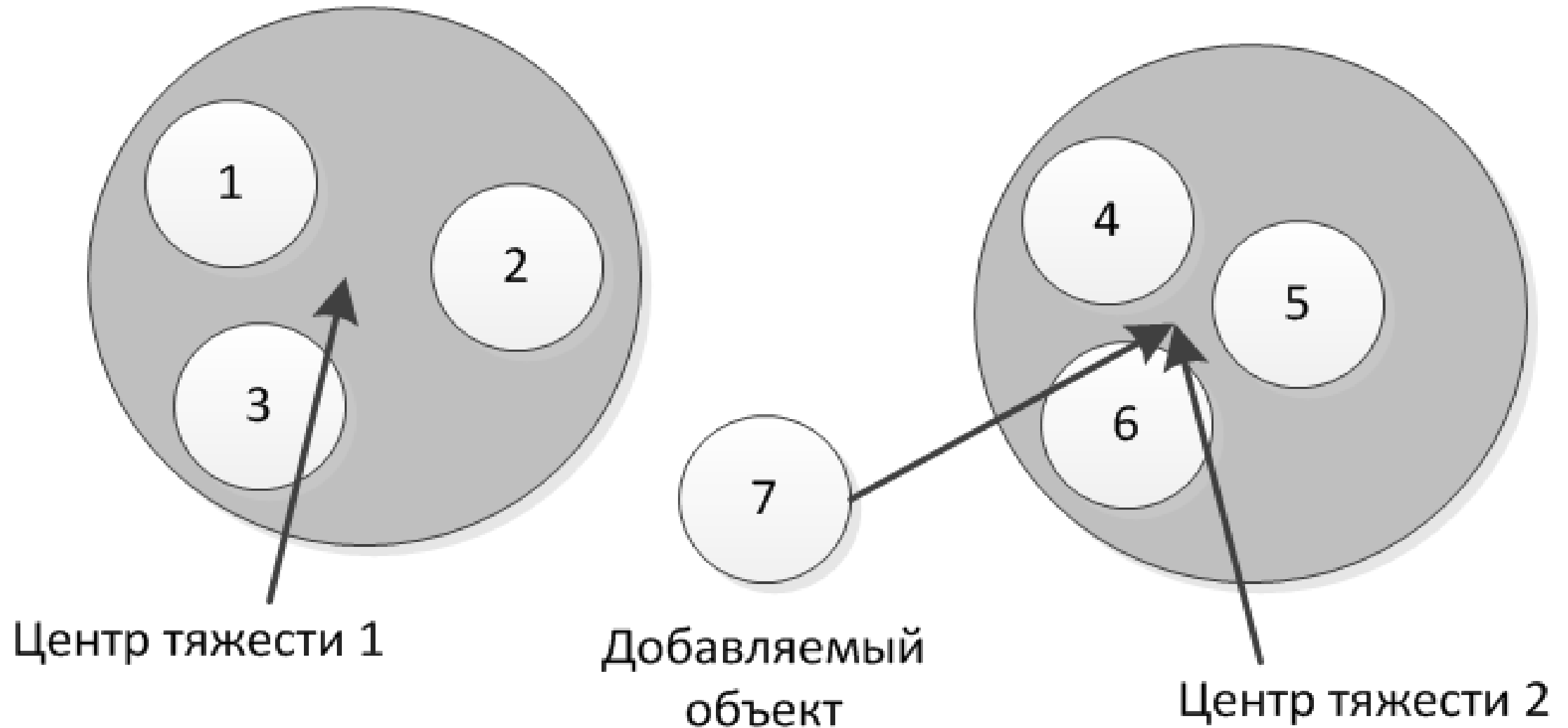
Самыми близкими объектами двух групп являются объекты 4 и 3. Расстояние до объекта 4 минимально.

Дальняя связь

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

В рассматриваемом примере на рисунке, представленном на следующем слайде, 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.

Включение объекта в группу объектов



Центр тяжести

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

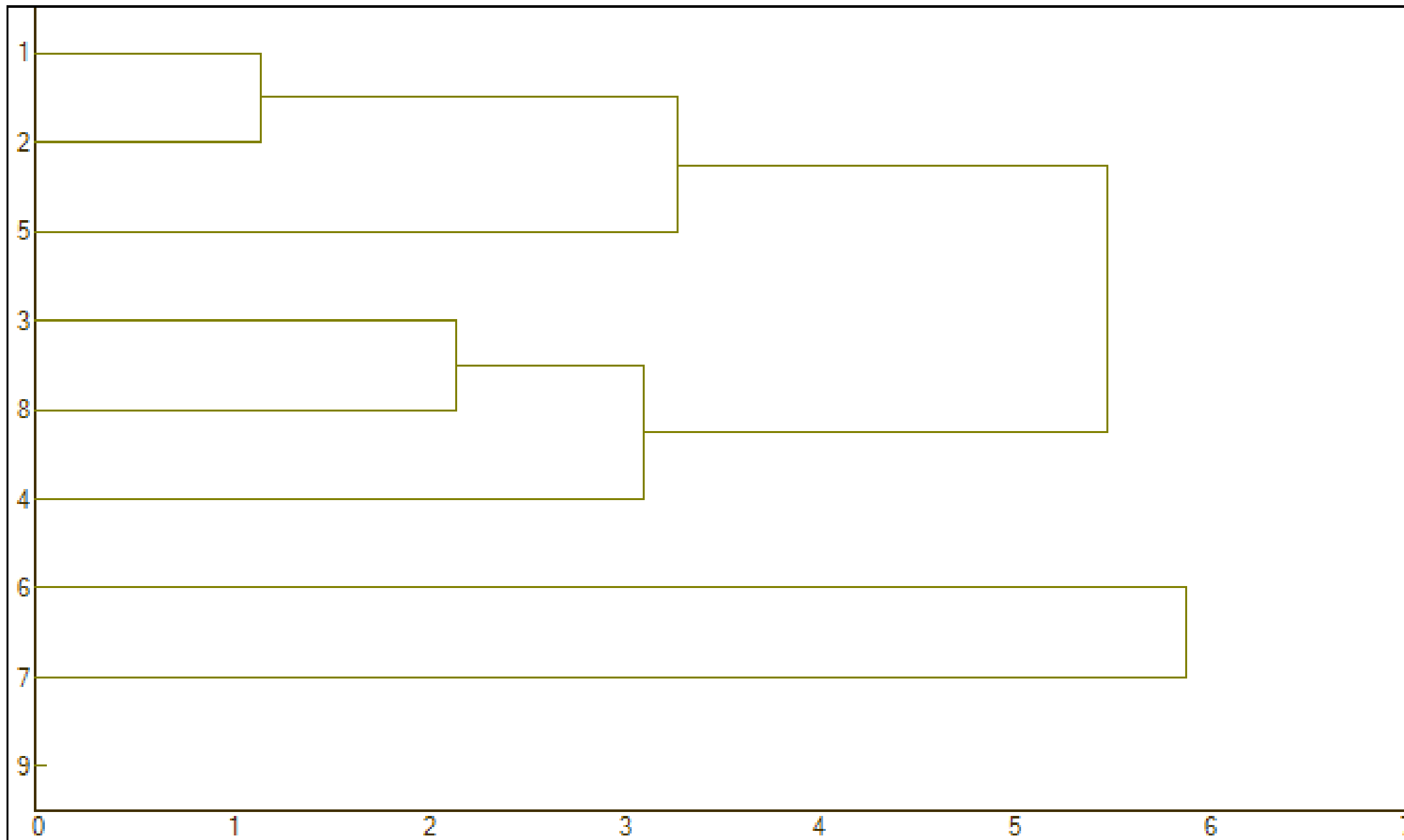
В примере, рассмотренном выше, объект 7 добавится в группу, содержащую объекты 4, 5 и 6. Расстояние до центра тяжести (некоего мифического объекта со средними значениями атрибутов) минимально.

k-средних

Пример расположения объектов



Дендрограмма связей



Цель кластеризации — выделение из множества объектов одной природы некоторого количества относительно однородных групп — сегментов или кластеров. 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.

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.

Возможные сценарии применения кластеризации на практике

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.

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.

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 predetermined данных конфигурации, функция ЗНАЧЕНИЕ().*

Два способа манипулирования данными в 1С

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.

Запросы предназначены для извлечения и обработки информации из базы данных для предоставления пользователю в требуемом виде. Под обработкой здесь подразумевается группировка полей, сортировка строк, расчет итогов и т.д. Изменять данные с помощью запросов в 1С нельзя

Общая схема работы с запросом

- ❑ Создание объекта Запрос и установка текста запроса;
- ❑ Установка параметров запроса;
- ❑ Выполнение запроса и получение результата;
- ❑ Обход результата запроса и обработка полученных данных.

Понятия, используемые, когда речь идет о запросах

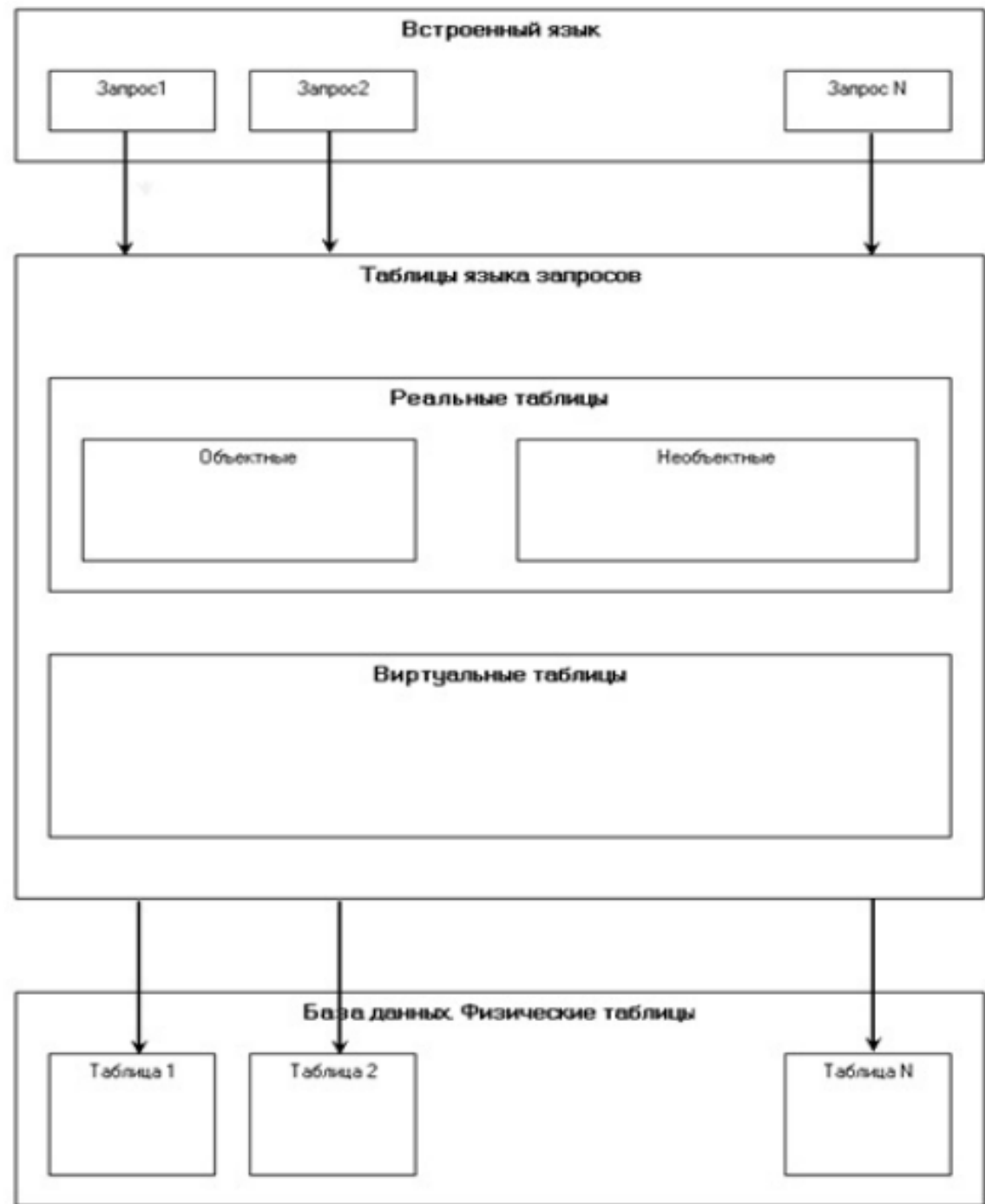
- ❑ Источники данных для запроса (таблицы системы, состав полей таблиц);
- ❑ Структура запроса, язык написания запроса;
- ❑ Порядок работы с запросом, получение данных, обход полученного результата.

Таблицы и поля базы данных

Логические таблицы в 1С:Предприятие» подразделяются на два основных класса:

- ❑ *Реальные таблицы* хранятся в БД. В случае использования реальной таблицы могут присутствовать вычисляемые поля, значения которых вычисляются как функция нескольких разных полей. Отдельный подкласс таблиц образуют так называемые объектные таблицы. Они предназначены для хранения состояния объектов системы, имеющих объектную природу. В таких таблицах 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. Виртуальная таблица может быть параметризована.

Доступ к данным в «1С:Предприятии»



В качестве поля таблицы может выступать:

□ **«Обычное поле»** (содержащее какое-либо значение, либо значение типа «Null»). Обычное поле отличается от вложенной таблицы тем, что в рамках одной записи обычному полю соответствует одно единственное значение

□ **Вложенная таблица.**
Вложенной таблице соответствует значение типа «РезультатЗапроса» с заранее заданным набором колонок.

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

Возможная структура запроса

<Описание запроса>

ВЫБРАТЬ [РАЗРЕШЕННЫЕ] [РАЗЛИЧНЫЕ] [ПЕРВЫЕ <Количество>]

<Список полей выборки>

[ИЗ <Список источников>]

[ГДЕ <Условие отбора>]

[СГРУППИРОВАТЬ ПО <Поля группировки>]

[ИМЕЮЩИЕ <Условие отбора>]

[ДЛЯ ИЗМЕНЕНИЯ [<Список таблиц верхнего уровня>]]

[<Объединение запросов>]

[<Упорядочивание результатов>]

[АВТОУПОРЯДОЧИВАНИЕ]

[<Описание итогов>]

Указание списка полей выборки

- ❑ После обязательного ключевого слова «Выбрать» следует *описание полей выборки*. Описание полей разделяются запятыми. После последнего описанного поля запятая не ставится.
- ❑ 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 и таблиц

- ❑ 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

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

// Данный пример демонстрирует использование

// в списке полей выборки псевдонима Товар,

// назначенного исходной таблице 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

Оператор ВЫБОР имеет конструкцию:

Код

ВЫБОР

КОГДА <Выражение> ТОГДА <Выражение>

ИНАЧЕ <Выражение>

КОНЕЦ

Дополнительные опции на закладке «Дополнительно»

Консоль запросов (тонкий клиент) * x Конструктор запроса x

Конструктор запроса

Таблицы и поля Группировка Условия **Дополнительно** Объединения/Псевдонимы Индекс Пакет запросов

Выборка записей Тип запроса

Первые Выборка данных Имя временной таблицы:

Без повторяющихся Создание временной таблицы

Разрешенные

Блокировать получаемые данные для последующего изменения

Таблицы

Таблицы для изменения

Валюты

Редактируемый запрос: Редактируемый оператор:

Запрос

Дополнительные опции на закладке «Дополнительно»

□ **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 задать предельное количество строк в результате запроса. Будут отображены самые первые (в соответствии с правилами упорядочивания результатов запроса) строки. Количество задается целым числом. В языке запросов добавлена возможность исполнения упорядочивания во вложенных запросах в случае, если вложенный запрос содержит конструкцию **ПЕРВЫЕ**.

// Необходимо отобрать пять самых дорогих товаров.

// Выборка должна осуществляться в порядке убывания цены товара.

Выбрать Первые 5

Справочник.Номенклатура.Наименование,

Справочник.Номенклатура.ЗакупочнаяЦена

Упорядочить По

Справочник.Номенклатура.ЗакупочнаяЦена Убыв

Дополнительные опции на закладке «Дополнительно»

- Без повторяющихся** – оставит в детальных записях выборки только уникальные записи. В тексте запроса сгенерит ключевое слово ВЫБРАТЬ РАЗЛИЧНЫЕ.

Пример:

// Необходимо узнать, каким вообще контрагентам

// отгружался товар за период.

Выбрать

Документ.РасходнаяНакладная.Контрагент

Дополнительные опции на закладке «Дополнительно»

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

Ключевое слово **РАЗРЕШЕННЫЕ** 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**.

Дополнительные опции на закладке «Дополнительно»

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**

Дополнительные опции на закладке «Дополнительно»

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.

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

Контрагенты.Наименование

ИЗ

Справочник.Контрагенты КАК Контрагенты

ГДЕ Контрагенты.Вид =

Значение (Перечисление.ВидыКонтрагентов.ЧастноеЛицо)

Условие отбора может определяться и как простое логическое выражение, и как более сложное, в котором простые логические выражения соединяются между собой логическими операторами И, Или, Не.

Выражение условия

- ❑ <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>... — разыменованное 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;
- ❑ <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 [A-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

- $\langle \text{реквизит дата} \rangle = \text{ДАТАВРЕМЯ}(\langle \text{реквизит года} \rangle, \langle \text{реквизит месяца} \rangle, \langle \text{реквизит числа} \rangle, \langle \text{реквизит часа} \rangle, \langle \text{реквизит минуты} \rangle, \langle \text{реквизит секунды} \rangle)$ —
- $\langle \text{реквизит} \rangle \text{ ЕСТЬ NULL}$ — выражение равно Истина, если значения NULL
- $\text{НЕ} (\langle \text{реквизит} \rangle \text{ ЕСТЬ NULL})$ — выражение равно Истина, если значение не NULL
- $\langle \text{псевдоним} \rangle . \langle \text{реквизит} \rangle = \text{ЗНАЧЕНИЕ} (\text{Справочник} . \langle \text{справочник} \rangle . \text{ПустаяСсылка})$ — сравнение с пустым значением
- $\langle \text{псевдоним} \rangle . \langle \text{реквизит} \rangle = \text{ЗНАЧЕНИЕ} (\text{Справочник} . \langle \text{справочник} \rangle . \langle \text{предопределенное значение} \rangle)$ — сравнение с предопределенным значением
- $\langle \text{псевдоним} \rangle . \langle \text{реквизит} \rangle = \text{ЗНАЧЕНИЕ} (\text{Перечисление} . \langle \text{имя перечисления} \rangle . \langle \text{значение перечисления} \rangle)$
- $\langle \text{псевдоним} \rangle . \langle \text{табличная часть} \rangle . \text{Ссылка}$ — значение ссылки в табличной части эквивалентно значению $\langle \text{псевдоним} \rangle . \text{Ссылка}$
- $\langle \text{ссылочный реквизит} \rangle . \text{Ссылка} = \&\langle \text{параметр условия} \rangle$ — 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 от 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](http://its.1c.ru)

если при **ЛЕВОЕ СОЕДИНЕНИЕ** задано условие к реквизиту правой таблицы (или наоборот), то фактически выполнится **ВНУТРЕННЕЕ СОЕДИНЕНИЕ**

Встроенные функции

ПОДСТРОКА (<Выражение>, <Значение>, <Значение>) |
ГОД (<Выражение>) |
КВАРТАЛ (<Выражение>) |
МЕСЯЦ (<Выражение>) |
ДЕНЬГОДА (<Выражение>) |
ДЕНЬ (<Выражение>) |
НЕДЕЛЯ (<Выражение>) |
ДЕНЬНЕДЕЛИ (<Выражение>) |
ЧАС (<Выражение>) |
МИНУТА (<Выражение>) |
СЕКУНДА (<Выражение>) |
НАЧАЛОПЕРИОДА (<Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада |
Полугодие) |
КОНЕЦПЕРИОДА (<Выражение>, Минута | Час | День | Неделя | Месяц | Квартал | Год | Декада |
Полугодие) |
ДОБАВИТЬКДАТЕ (<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 ()
СГРУППИРОВАНАПО (<Выражение>)

Задание в запросе значений predetermined данных конфигурации, функция ЗНАЧЕНИЕ()

Обращение в запросах к predetermined данным конфигурации и значениям системных перечислений осуществляется с помощью литерала функционального типа [1]:

ЗНАЧЕНИЕ (<ПредставлениеЗначения>)

Для системных перечислений представление значения имеет вид:

<ИмяСистемногоПеречисления> . <Значение>

Для predetermined данных конфигурации представление значения имеет вид:

<ТипПредeterminedЗначения> . <ИмяОбъектаМетаданных> . <Значение>

Тип predetermined значения может быть:

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

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.

ВЫБРАТЬ

Накладная.Номенклатура,

СУММА (Накладная.Количество) **КАК** Количество

ИЗ

Документ.РасходнаяНакладная.Состав **КАК** Накладная

Сгруппировать **По**

Накладная.Номенклатура

ИМЕЮЩИЕ

СУММА (Накладная.Количество) **>** 20

Агрегатные функции в языке запросов

СУММА (<Выражение>) |

СРЕДНЕЕ (<Выражение>) |

МИНИМУМ (<Выражение>) |

МАКСИМУМ (<Выражение>) |

КОЛИЧЕСТВО ([РАЗЛИЧНЫЕ] <Выражение> | *)

Конструкция ИМЕЮЩИЕ (HAVING)

Позволяет указать отбор в виде агрегатной функции, этого нельзя сделать в конструкции ГДЕ.

Пример: отберем количество товаров, которых поступило более 5 штук

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

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

РАЗЛИЧНЫЕ

Означает, что в результат не попадет дублирующих строк

ПЕРВЫЕ n

Запрос отберет только n первых записей

ЕСТЬNULL (ISNULL)

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 NULLи поставить понятный 0 (ноль).

ЕСТЬNULL(НалогиПредМесяца.ПримененнаяЛьготаФСС, 0)

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.

Разыменование полей

Обращение к полям через точку называется операцией разыменования ссылочного поля. Пример:

Оплата.Организация.АдминистративнаяЕдиница

Запрос:

ВЫБРАТЬ Оплата.Ссылка, Оплата.Организация,
Оплата.Организация.АдминистративнаяЕдиница
ИЗ Документ.Оплата КАК Оплата

Можно представить в виде:

ВЫБРАТЬ Оплата.Ссылка, Оплата.Организация, Оплата.Организация,
Организации. АдминистративнаяЕдиница
ИЗ Документ.Оплата КАК Оплата ЛЕВОЕ СОЕДИНЕНИЕ
Справочник.Организации КАК Организации
ПО Оплата.Организация = Организации.Ссылка

Конструкция ВЫРАЗИТЬ()

При разыменовании ссылочных полей составного типа платформа пытается создать неявные соединения со всеми таблицами, которые входят в тип этого поля. В этом случае запрос будет неоптимален. Если четко известно, какого типа поле, необходимо ограничивать такие поля по типу конструкцией **ВЫРАЗИТЬ()**

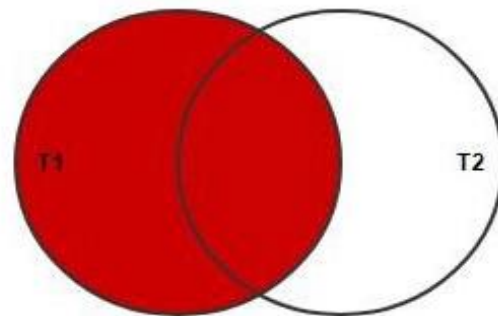
ВЫБРАТЬ

**ВЫРАЗИТЬ(НераспределенныеОплаты.Регистратор
КАК Документ.Оплата).Дата,**

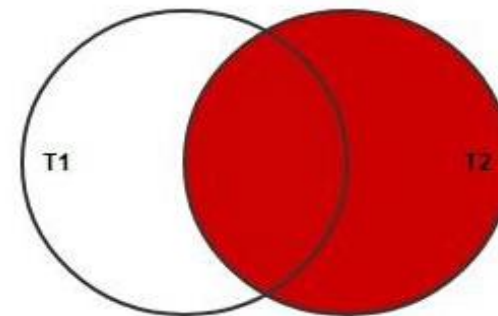
**ИЗ РегистрНакопления.НераспределенныеОплаты
КАК НераспределенныеОплаты**

Виды соединений

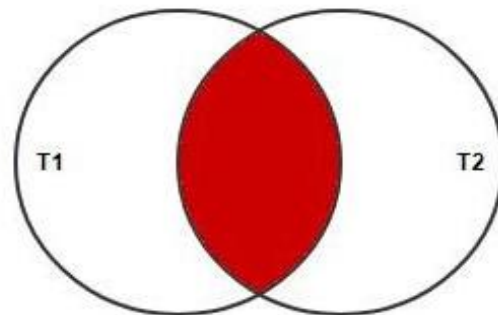
- внутреннее соединение;
- левое внешнее соединение;
- правое внешнее соединение;
- полное внешнее соединение.



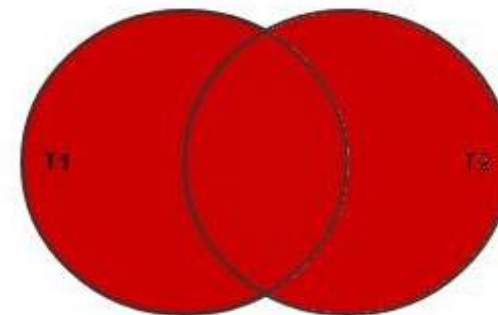
ЛЕВОЕ СОЕДИНЕНИЕ



ПРАВОЕ СОЕДИНЕНИЕ



ВНУТРЕННЕЕ СОЕДИНЕНИЕ



ПОЛНОЕ 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 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

ВЫБРАТЬ

СоставЗаказа.Товар,

СоставЗаказа.Количество,

СоставЗаказа.Сумма

ИЗ

Документ.ЗаказТовара.Состав КАК СоставЗаказа

Поле Ссылка таблицы, представляющей табличную часть

Чтобы получить данные из основной таблицы документа, нужно обращаться к полям таблицы через точку от поля табличной части Ссылка

ВЫБРАТЬ

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

ИЗ

Документ.ЗаказТовара.Состав КАК СоставЗаказа

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.

Выбрать
Константы.Адрес,
Константы.ДатаУчрежденияКомпании

Выбрать
Адрес, ДатаБлокировки
Из
Константы

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 структура, определяемая ключевым словом «Где <Условие отбора>». Условие действует на момент получения данных запросом.

Примеры:

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

Выбрать

Код, Наименование

Из

Справочник.Номенклатура

Где ЭтоГруппа

- ❑ В условиях можно использовать стандартные логические операции

ЭтоГруппа=Ложь И (Не ПометкаУдаления)

- ❑ Выбрать

Код, Наименование

Из

Справочник.Номенклатура

У 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:

- Внутреннее соединения
- Левое внешнее соединение
- Правое внешнее соединение
- Полное внешнее соединение.

Примеры:

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

В соответствии с условием можно выделить записи, для которых условие выполняется:

ИЛ 86	AA1
-------	-----

Ty 154	AA2
--------	-----

AA1	Петров
-----	--------

AA1	Чемисов
-----	---------

AA2	Вольский
-----	----------

Записи, не удовлетворяющие условию соединения:

Из таблицы №1

ИЛ 62	AA3
-------	-----

?	Null
---	------

Из таблицы №2

Null	?
------	---

AA4	Третьяков
-----	-----------

Варианты соединений

Внутреннее соединение

ИЛ 86	Петров
ИЛ 86	Чемисов
Ту 154	Вольский

Левое внешнее соединение

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

Правое внешнее соединение

ИЛ 86	Петров
ИЛ 86	Чемисов
Ту 154	Вольский
Null	Третьяков

Полное внешнее соединение

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. В качестве 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=&Номен

...

Запрос.УстановитьПараметр(«Номен», ТекТовар);

Выполнение и работа с запросами во встроенном языке

Для формирования запросов, выборки и обработки результатов запросов в языке предусмотрен специальный набор объектов. С помощью этих объектов выполняется формирование запроса, обход записей запроса и т. д. 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**.

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.
Выборка = РезультатЗапроса.Выбрать ();
// Пока в выборке есть записи ...
Пока Выборка.Следующий () Цикл
    // ... выведем в окно сообщений поля из результата.
    Товар = Выборка.Наименование;
    Родитель = Выборка.НаименованиеРодителя;
    Сообщить ("Товар: " + Товар + " Родитель: " + Родитель);
КонецЦикла;
```

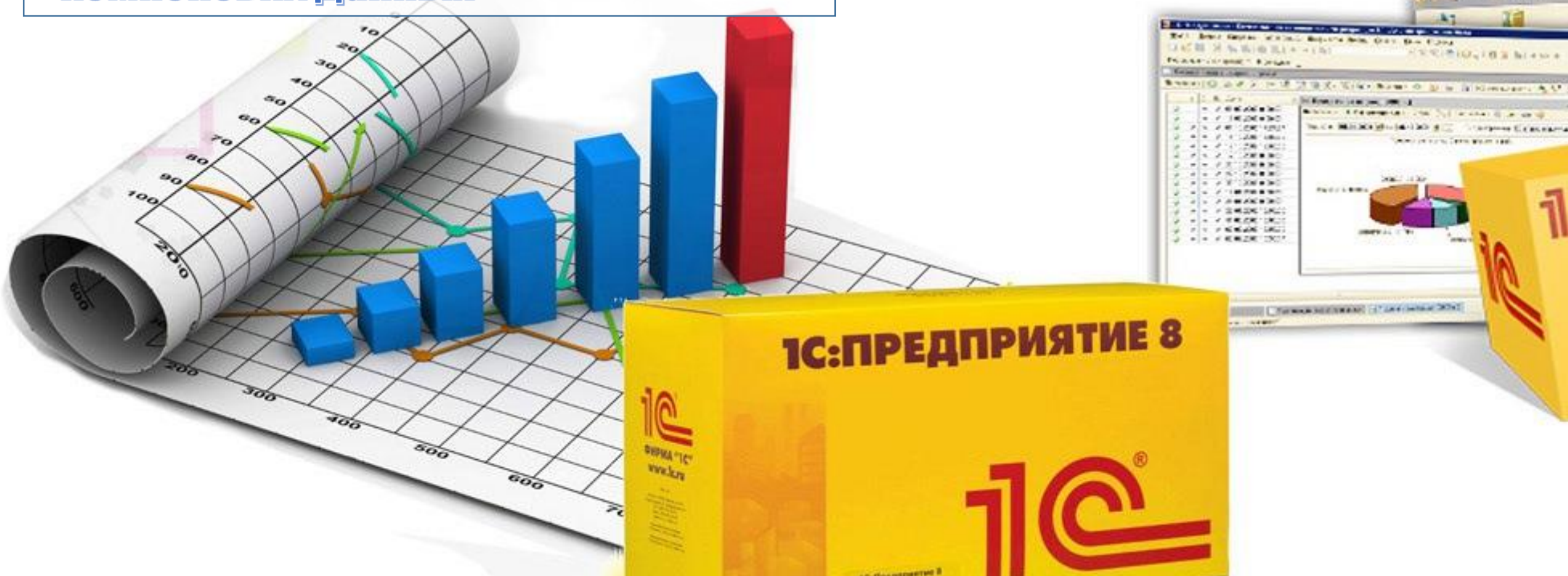
Пример сложного запроса

```
Запрос = Новый Запрос;  
Запрос.Текст =  
    "ВЫБРАТЬ  
    |   ПродажаТоваровТовары.Номенклатура КАК Номенклатура,  
    |   СУММА (ПродажаТоваровТовары.Количество) КАК Количество,  
    |   ПродажаТоваровТовары.Ссылка.Склад КАК Склад  
    |ПОМЕСТИТЬ ТабДок  
    |ИЗ  
    |   Документ.ПродажаТоваров.Товары КАК ПродажаТоваровТовары  
    |ГДЕ  
    |   ПродажаТоваровТовары.Ссылка = &Ссылка  
    |   И ПродажаТоваровТовары.Номенклатура.ВидНоменклатуры <> ЗНАЧЕНИЕ (Перечисление.ВидыТоваров.Услуга)  
    |  
    |СГРУППИРОВАТЬ ПО  
    |   ПродажаТоваровТовары.Номенклатура,  
    |   ПродажаТоваровТовары.Ссылка.Склад  
    |;  
    |  
    |////////////////////////////////////
```

Пример сложного запроса

```
//////  
| ВЫБРАТЬ  
|   ТабДок.Номенклатура КАК Номенклатура,  
|   ТабДок.Количество КАК Количество,  
|   ТабДок.Склад КАК Склад,  
|   ЕСТЬNULL (ОстаткиНоменклатурыОстатки.КоличествоОстаток, 0) КАК КоличествоОстаток,  
|   ЕСТЬNULL (ОстаткиНоменклатурыОстатки.СуммаОстаток, 0) КАК СуммаОстаток,  
|   ОстаткиНоменклатурыОстатки.Партия КАК Партия,  
|   ЕСТЬNULL (БронированиеТоваровОстатки.КоличествоЗабронированногоОстаток, 0) КАК КоличествоЗабронированного,  
|   ОстаткиНоменклатурыОстатки.Партия.МоментВремени КАК ПартияДата  
| ИЗ  
|   ТабДок КАК ТабДок  
|     ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.ОстаткиНоменклатуры.Остатки (  
|       &Момент,  
|       (Номенклатура, Склад) В  
|         (ВЫБРАТЬ  
|           ТабДок.Номенклатура,  
|           ТабДок.Склад  
|         ИЗ  
|           ТабДок КАК ТабДок) ) КАК ОстаткиНоменклатурыОстатки  
|   ПО ТабДок.Номенклатура = ОстаткиНоменклатурыОстатки.Номенклатура  
|     ЛЕВОЕ СОЕДИНЕНИЕ РегистрНакопления.БронированиеТоваров.Остатки (  
|       &Момент,  
|       (Номенклатура, Склад) В  
|         (ВЫБРАТЬ  
|           ТабДок.Номенклатура,  
|           ТабДок.Склад  
|         ИЗ  
|           ТабДок КАК ТабДок) ) КАК БронированиеТоваровОстатки  
|   ПО ТабДок.Номенклатура = БронированиеТоваровОстатки.Номенклатура  
| УПОРЯДОЧИТЬ ПО  
|   Номенклатура,  
|   ПартияДата,  
|   Партия  
| ИТОГИ  
|   МАКСИМУМ (Количество) ,  
|   СУММА (КоличествоОстаток) ,  
|   СУММА (СуммаОстаток) ,  
|   МАКСИМУМ (КоличествоЗабронированного)  
| ПО  
|   Номенклатура";
```


**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)

Это мощный механизм платформы «1С:Предприятие» и новее) для разработки отчетов.

была придумана 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.

Использование декларативного описания отчетов позволяет реализовать следующие возможности

- создание отчета без программирования,
- возможность создания различных вариантов отчета,
- возможность задания различных вариантов пользовательских настроек,
- использование автоматически генерируемых форм просмотра и настройки отчета,
- разбиение исполнения отчета на этапы,
- исполнение отдельных этапов построения отчета на различных компьютерах,
- независимое использование отдельных частей системы компоновки данных,
- программное влияние на процесс выполнения отчета,
- настройки структуры отчета,
- совмещение в отчете нескольких таблиц,
- создание вложенных отчетов и др.

- 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.
- Формирование отчета с использованием системы компоновки данных разбивается на несколько *этапов*.
- При этом разбиение на этапы – не логическое, а *физическое*. То есть в каждый из этих этапов разработчик может вмешаться, программно или визуально, выполнить различные этапы компоновки данных в различных модулях и даже на различных компьютерах

Схема взаимодействия объектов системы компоновки



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.

Схема разработки и использования отчета с помощью СИСТЕМЫ КОМПОНОВКИ ДАННЫХ

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.

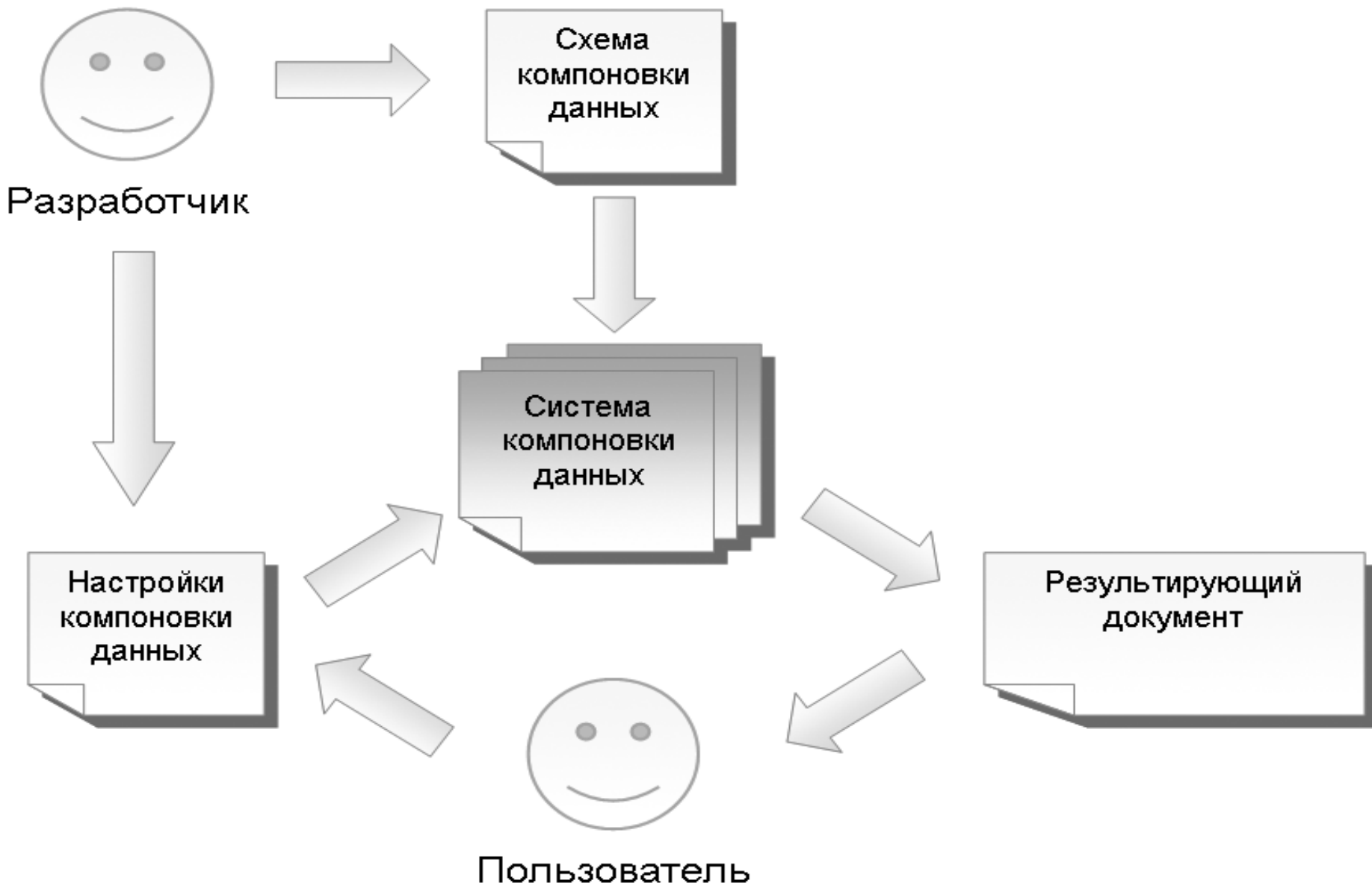


Схема компоновки данных

Основой для компоновки отчета является схема компоновки данных.

Создание схемы компоновки данных может быть выполнено:

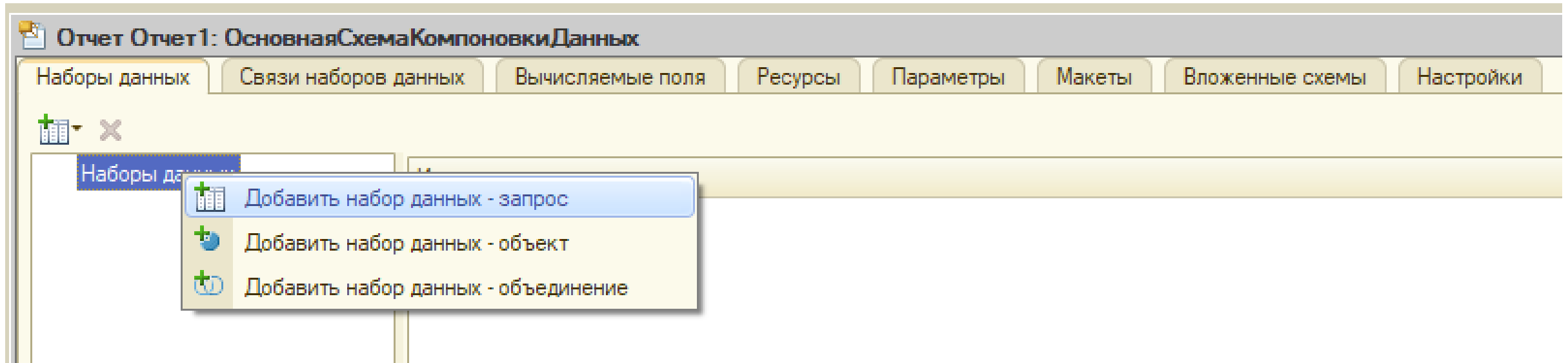
- визуально при помощи конструктора схемы компоновки данных;
- визуально при помощи любого редактора, позволяющего редактировать текст XML;
- программно при помощи объектов встроенного языка системы «1С:Предприятие».

Схема компоновки данных содержит следующие разделы:

- наборы данных,
- связи наборов данных,
- вычисляемые поля,
- ресурсы,
- параметры,
- макеты,
- вложенные схемы,
- настройки.

Наборы данных

Наборы данных являются источниками данных для отчета и описывают способ получения этих данных. Данные можно получить из информационной базы или из некоторых объектов, находящихся в памяти (например, из таблицы значений). Один отчет может использовать одновременно несколько наборов данных, отбирая, объединяя и связывая между собой данные из разных наборов.



Виды наборов данных

Существуют различные виды наборов данных. Они отличаются тем, откуда получаются данные и как эти данные обрабатываются:

- ❑ **Запрос** – содержит запрос к базе данных на языке запросов с использованием дополнительных синтаксических элементов (расширений);
- ❑ **Объект** – содержит информацию о внешнем наборе данных, например, таблице значений;
- ❑ **Объединение** – объединяет информацию из нескольких наборов данных.

- ❑ Разработчик, создавая схему компоновки данных, может добавить в нее наборы данных такого вида и в таком количестве, которые ему необходимы. В зависимости от вида набора данных нужно выполнить различные действия для того, чтобы отчет мог использовать данные этого набора. Например, если используется набор данных – запрос к информационной базе «1С:Предприятия».
- ❑ Если используется набор данных – объект, то в схеме компоновки нужно описать поля этого набора данных, чтобы система компоновки знала, данные какого типа будут получены из этого набора, затем с помощью встроенного языка создать внешний набор данных и инициализировать им процессор компоновки данных.
- ❑ Если используется набор данных – объединение, то в схеме компоновки данных нужно описать наборы данных, которые входят в это объединение. В схеме компоновки данных нужно задать текст запроса, обращающегося к информационной базе «1С:Предприятия».
- ❑ Если используется набор данных – объект, то в схеме компоновки нужно описать поля этого набора данных, чтобы система компоновки знала, данные какого типа будут получены из этого набора, затем с помощью встроенного языка создать внешний набор данных и инициализировать им процессор компоновки данных. Если используется набор данных – объединение, то в схеме компоновки данных нужно описать наборы данных, которые входят в это объединение.

Связи наборов данных

Схема компоновки данных может содержать несколько наборов данных. В случае, когда требуется получить связанную информацию из этих наборов, в схеме компоновки данных описывается их связь по какому-то полю. При связывании наборов данных в схеме компоновки используется Левое соединение.

Однако в случае, если требуется вывести в отчет все записи из обоих наборов данных, используется набор данных – объединение. Такой набор содержит два или более подчиненных набора данных, но информация в них не связывается, а объединяется. По аналогии с языком запросов в этом случае используется Объединение.

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.

Продажа системных блоков

Параметры: Начало периода: 01.12.2011
Конец периода: 31.12.2011

Склад	Количество Приход	Количество Расход	Конечный остаток	Стоимость Приход	Стоимость Расход	Прибыль
Системный блок						
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

Ресурсы

Система компоновки данных может рассчитывать и добавлять в отчет записи, которые называются итоговыми. Этих записей в явном виде нет в базе данных, но они получаются в результате исполнения отчета, на основании некоторой обработки детальных записей, которую выполняет система компоновки данных.

Эти групповые
итоговые данные
называются
ресурсами схемы
компоновки данных.

Остатки номенклатуры

Склад	Начальный остаток	Приход	Расход	Конечный остаток
Номенклатура				
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

- ❑ Обычно данные в отчете группируются по каким-то позициям, например по номенклатуре.
- ❑ Смысл группировки не только в определенном расположении данных, но и в выводе итоговых данных – ресурсов по этой группировке.
- ❑ На основании значений ресурсов формируются общие итоги отчета.
- ❑ Для вывода таблиц и диаграмм обязательно определять ресурсы для отчета.
- ❑ Отчет, выводящий данные в группировку, может и не содержать ресурсов.

Вывод ресурсов в таблице

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

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.
- ❑ Как правило, параметры выводятся пользователю перед формированием отчета. Затем заданные пользователем значения параметров передаются в отчет, и отчет формируется заново, например, с новым отчетным периодом.

Вывод в отчете параметров отчетного периода

Отчет по системным блокам - Демонстрационная к... (1С:Предприятие)

Отчет по системным блокам

Вариант отчета:

Все действия ▾

<input checked="" type="checkbox"/>	Начало периода	01.12.2011
<input checked="" type="checkbox"/>	Конец периода	31.12.2011

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

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 predetermined макеты. В отличие от стандартных макетов оформления, predetermined макеты привязаны к одному отчету, его конкретным полям и структуре.
- ❑ Можно создавать макеты отдельных полей, группировок, итоговых полей и ресурсов отчета. В приведенном примере группировка по номенклатуре оформлена с помощью predetermined макета.

Макеты

Контрагент ООО "Автоматика"

Контрагент	Стоимость	Стоимость.% в группировке	Стоимость.% общий
Период, месяц			
Период, день			
(000000001) ООО "Автоматика"	24 500,00	100,00	100,00
Декабрь.2011	11 500,00	46,94	46,94
05.12.2011	1 000,00	8,70	4,08
11.12.2011	10 500,00	91,30	42,86
Январь.2012	13 000,00	53,06	53,06
12.01.2012	13 000,00	100,00	53,06
Итого	24 500,00	100,00	100,00

Контрагент ООО "Атлант"

Контрагент	Стоимость	Стоимость.% в группировке	Стоимость.% общий
Период, месяц			
Период, день			
(000000005) ООО "Атлант"	9 000,00	100,00	100,00
Январь.2012	9 000,00	100,00	100,00
13.01.2012	4 000,00	44,44	44,44
15.01.2012	5 000,00	55,56	55,56
Итого	9 000,00	100,00	100,00

Контрагент ООО "Система"

Контрагент	Стоимость	Стоимость.% в группировке	Стоимость.% общий
Период, месяц			
Период, день			
(000000004) ООО "Система"	10 000,00	100,00	100,00
Декабрь.2011	10 000,00	100,00	100,00
09.12.2011	10 000,00	100,00	100,00
Итого	10 000,00	100,00	100,00

Вложенные схемы

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. В приведенном примере выводится список контрагентов с вложенными данными по продажам номенклатуры для каждого из них.

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.

Параметры схемы компоновки (продолжение)

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 функциональной опции.

Значения не параметризуемых функциональных опций также учитываются 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 значений параметра.

Использование – определяется использование параметра в итоговом запроса 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

**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*

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

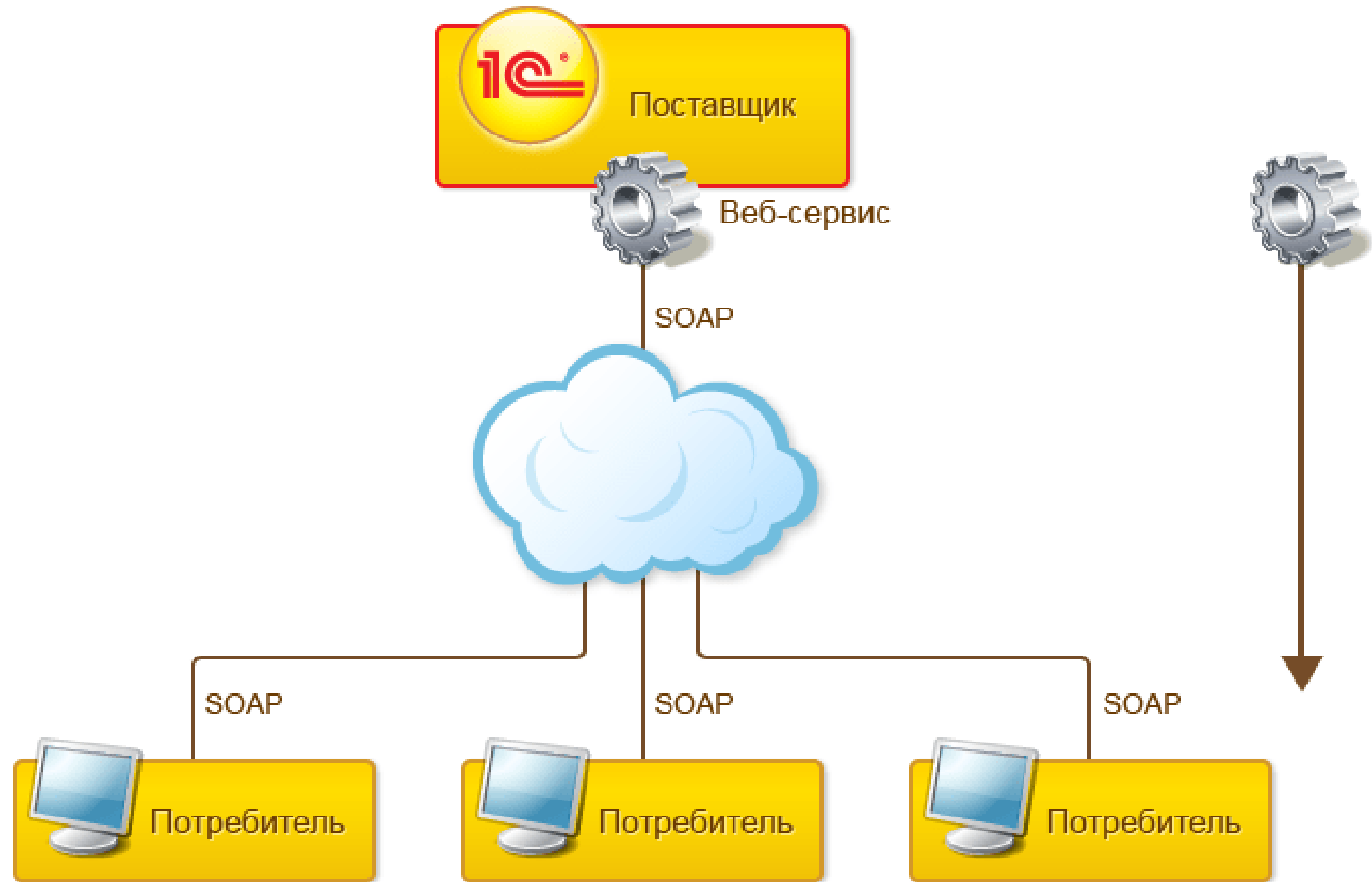
данными могут быть условно разделены на два уровня:

- ❑ универсальные механизмы обмена данными,
- ❑ распределенные информационные базы.

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.

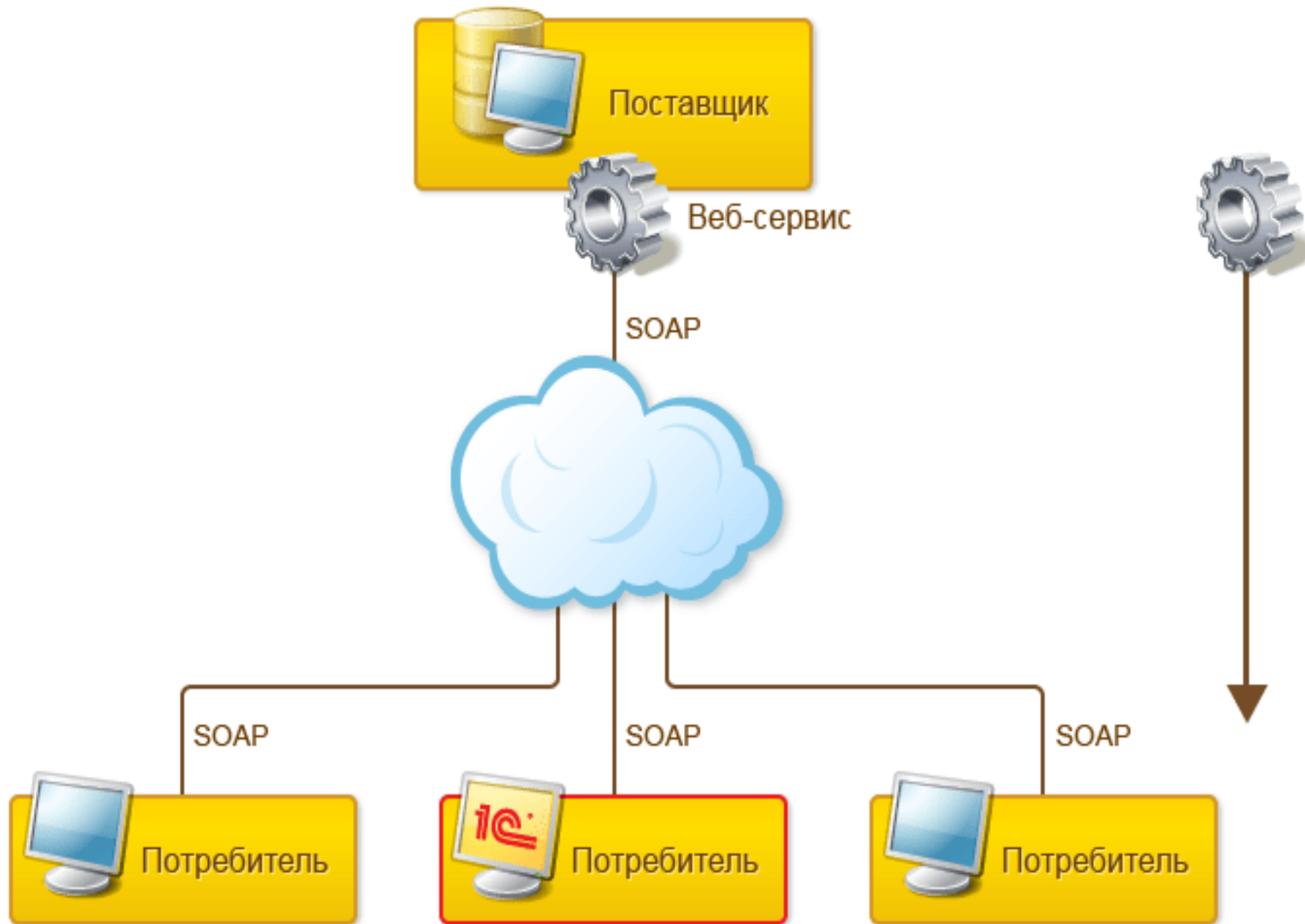
«1С:Предприятие» — поставщик веб-сервисов



«1С:Предприятие» — поставщик веб-сервисов

- ❑ 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



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.

WS-ссылка

Конфигурация

Действия

Поиск (Ctrl+Alt+M)

- Группы команд
- Общие макеты
- Общие картинки
- XDTO-пакеты
- Web-сервисы
- HTTP-сервисы
- WS-ссылки**
- ДанныеНоменклатуры**
- Элементы стиля
- Языки
- Константы
- Справочники
- Документы
- Журналы документов
- Перечисления
- Отчеты
- Обработки
- Планы видов характеристик
- Планы счетов
- Планы видов расчета
- Регистры сведений
- Регистры накопления
- Регистры бухгалтерии
- Регистры расчета
- Бизнес-процессы
- Задачи
- Внешние источники данных

WS-сс...: WS-ссылка [Только для чтения]

- WS-ссылка
 - Модель данных
 - Пакет (<http://www.1c.ru/demos/products>)
 - Свойства
 - Получить Response
 - Получить
 - Типы объектов
 - Номенклатура
 - Наименование
 - ЗакупочнаяЦена
 - Картинка
 - ПолноеНаименование
 - ШтрихКод
 - НоменклатураГруппа
 - Наименование
 - Группы
 - Элементы
 - Получить
 - Получить Response
 - return
 - Web-сервисы
 - ДанныеНоменклатуры
 - ДанныеНоменклатурыSoap
 - Получить
 - ДанныеНоменклатурыSoap12
 - Получить

Динамическое обращение к веб-сервисам

- ❑ 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.
- ❑ 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.).

Пример создания прокси веб-сервиса и использования одного из методов веб-сервиса

```
// 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);
```

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

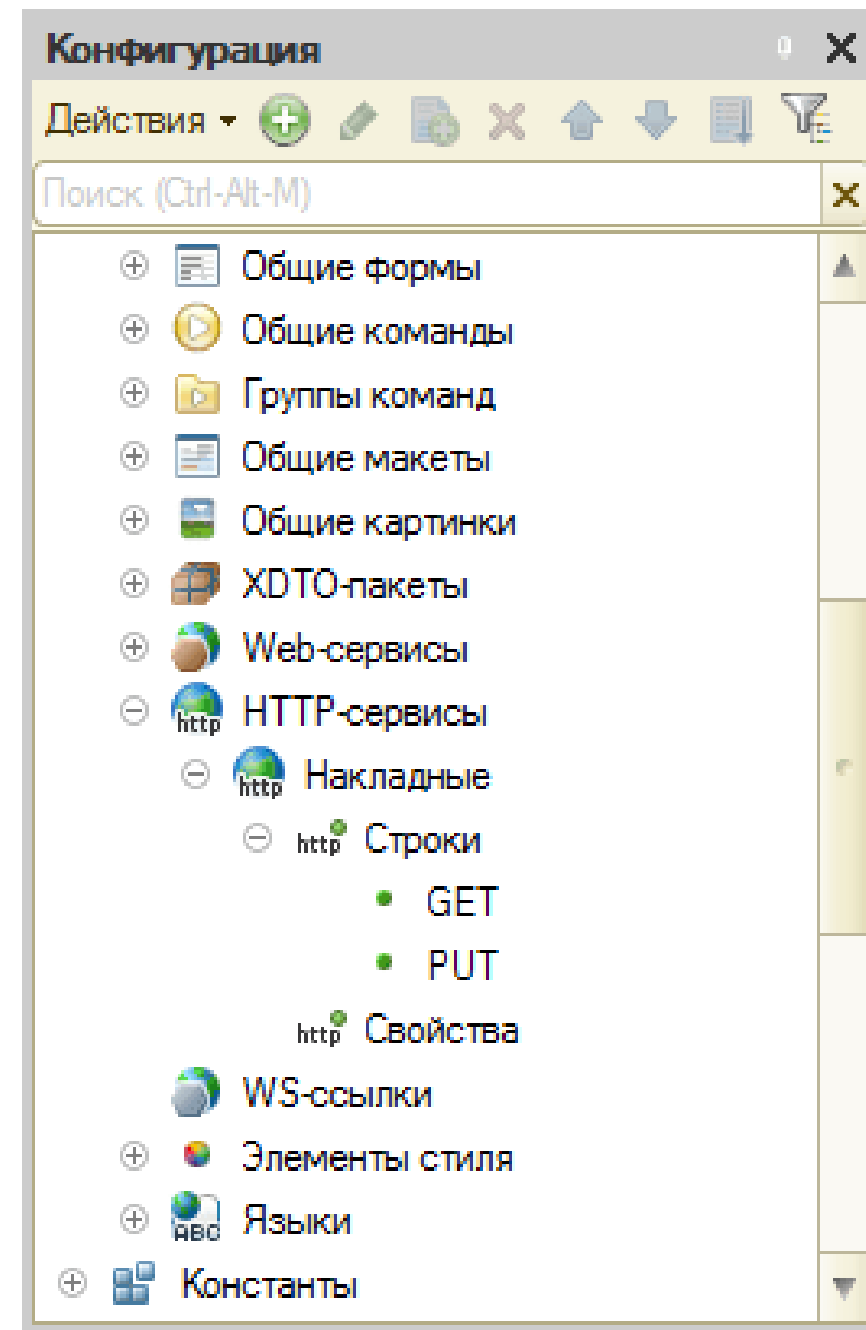
Использование HTTP-сервисов в 1С



HTTP-сервисы

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.



Шаблоны HTTP-сервисов

HTTP-сервис Накладные

Основные
Подсистемы
Шаблоны URL
Прочее

http

- Строки
 - GET
 - PUT
- Свойства

Свойства: Строки

Поиск (Ctrl-Alt-I)

Основные:

Имя: Строки

Синоним: Строки

Комментарий

Шаблон: /{Номер}/Строки/{НомерСтроки}

Свойства: GET

Поиск (Ctrl-Alt-I)

Основные:

Имя: GET

Синоним: GET

Комментарий

HTTP-метод: GET

Обработчик: СтрокиGET

Действия <Назад Далее> Закрывать Справка

Метод HTTP-сервиса

```
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];

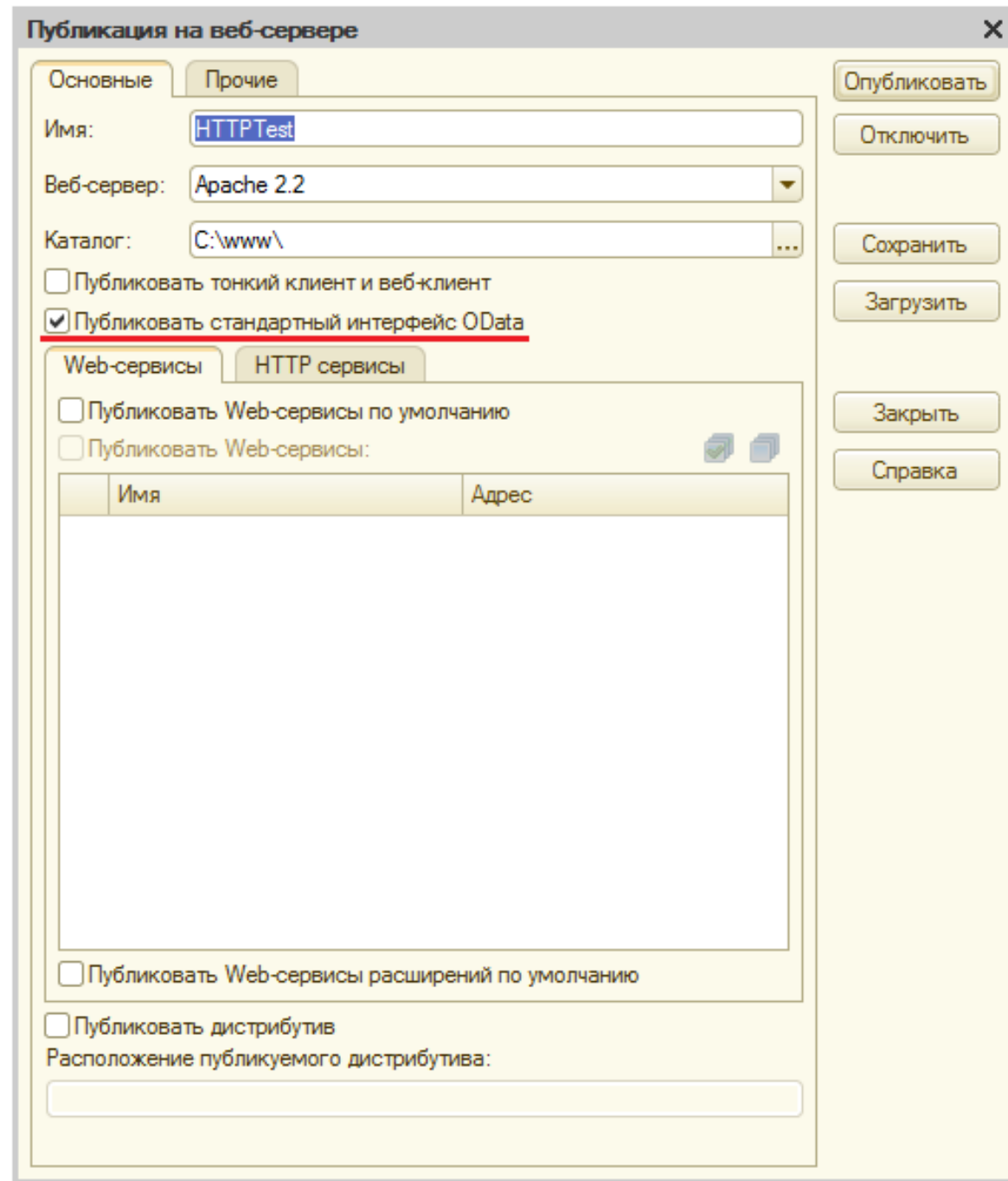
    // Преобразовать данные строки в XML.
    ЗаписьXML = Новый ЗаписьXML;
    ЗаписьXML.УстановитьСтроку ();
    ЗаписьXML.ЗаписатьОбъявлениеXML ();
    ЗаписьXML.ЗаписатьНачалоЭлемента ("answer");
    ЗаписьXML.ЗаписатьНачалоЭлемента ("СтрокаНакладной");
    ЗаписьXML.ЗаписатьАтрибут ("Товар",      СтрокаДокумента.Товар.Наименование);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

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

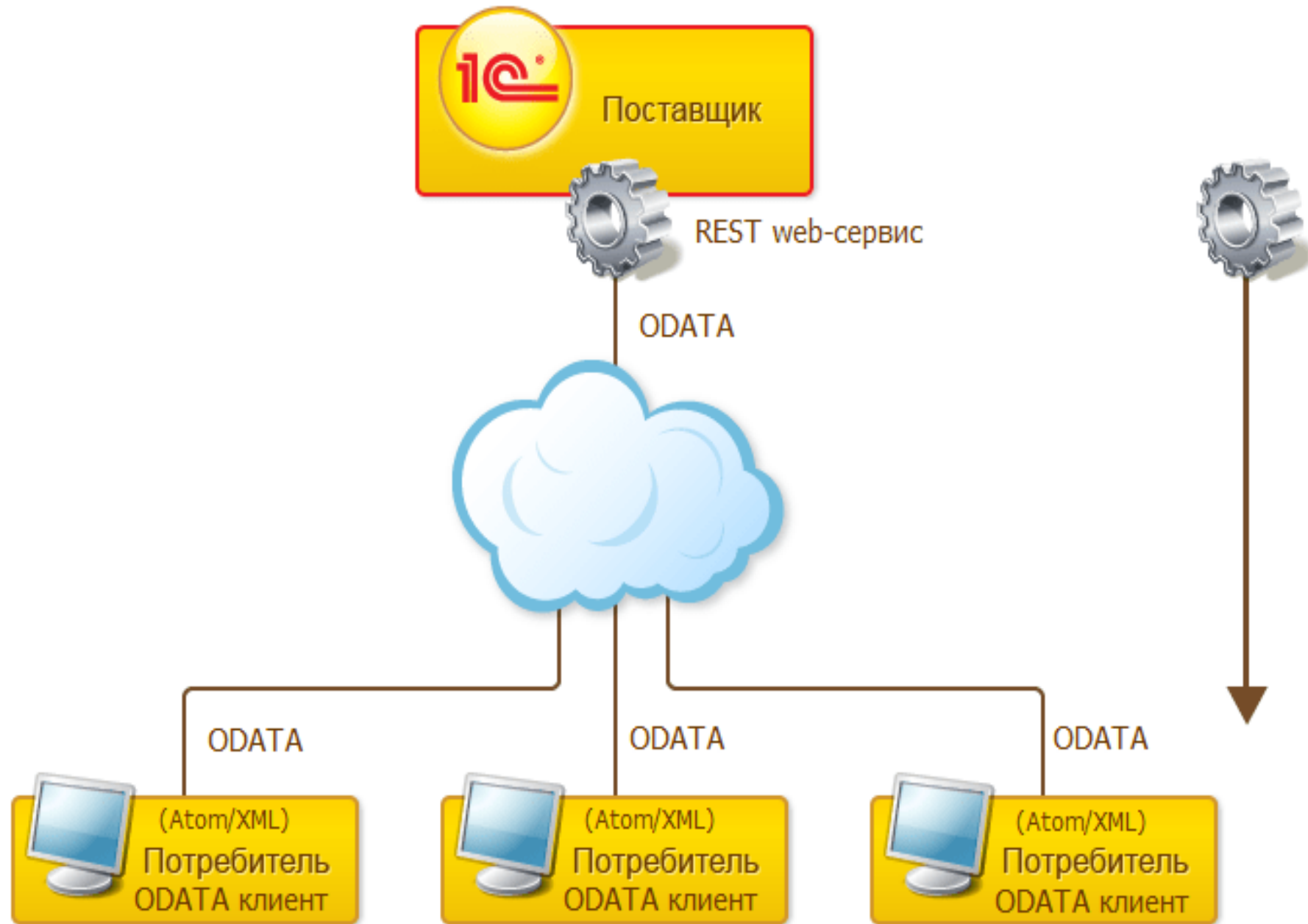
    Возврат Ответ;

Конецфункции
```

Поддержка REST-интерфейса в 1С



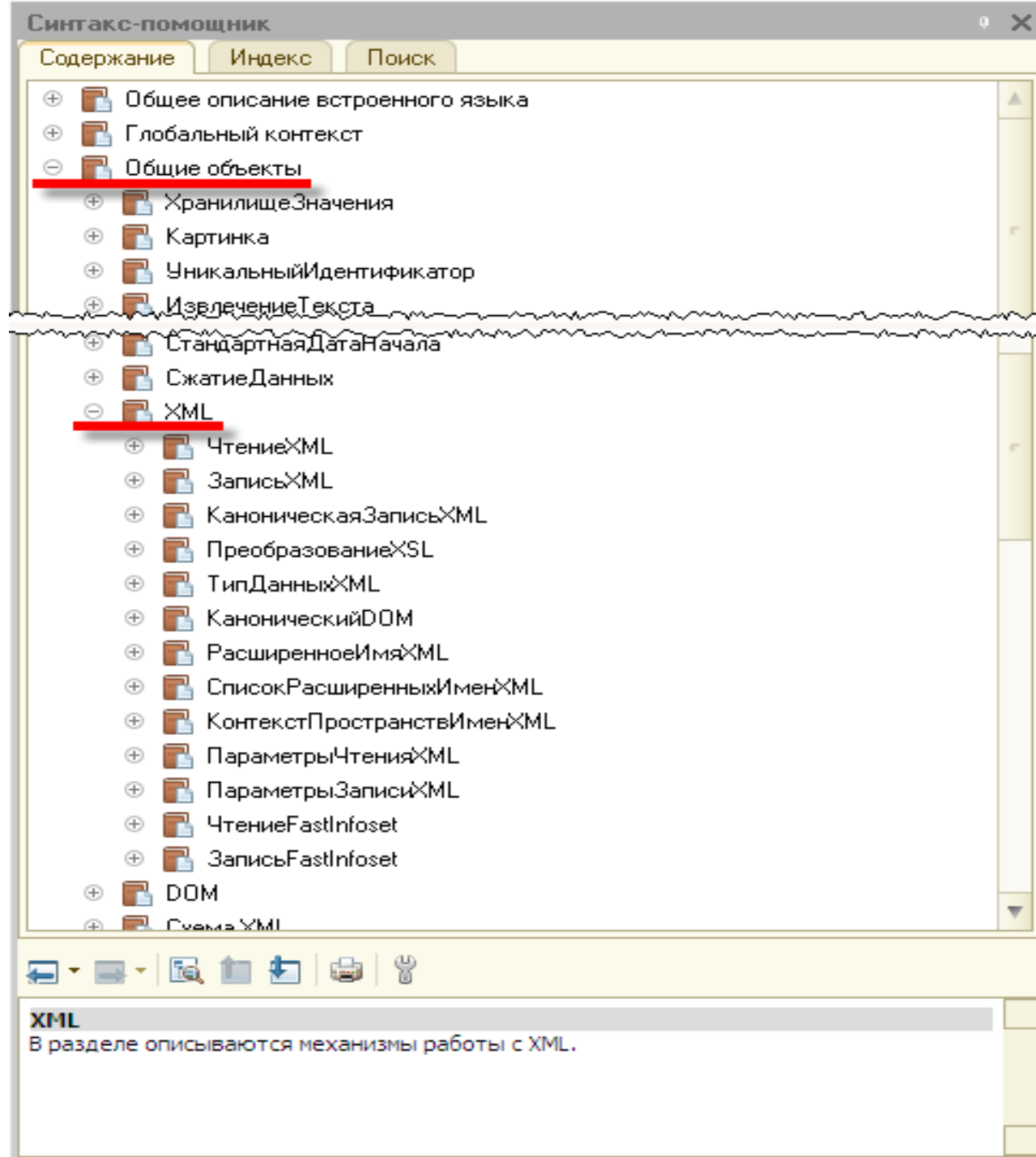
REST web-сервис



Использование стандартного интерфейса OData

- ❑ 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.

Применение формата XML в 1С



Пример схемы использования для интеграция BizTalk сервера компании Microsoft



Пример листинга ПОТОКОВОЙ записи JSON

```
// 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();
```

Результат записи JSON

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


Пример сериализации (записи) в JSON

```
// 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.Закрыть ();
```

```
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 }
```

Код, выполняющий сериализацию элемента справочника

```
// 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();
```

Результат сериализации элемента справочника

```
1  {
2    "#type": "jcfg: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@hleb.ru",
18     "Факс": "",
19     "ВебСайт": "",
20     "ВидЦен": "9d5c4225-8c4c-11db-a9b0-00055d49b45e",
21     "ДополнительнаяИнформация": "Поставка хлеба",
22     "КонтактноеЛицо": "Громышева П.Р.",
23     "Широта": 55.762744,
24     "Долгота": 37.618102
25   }
26 }
```

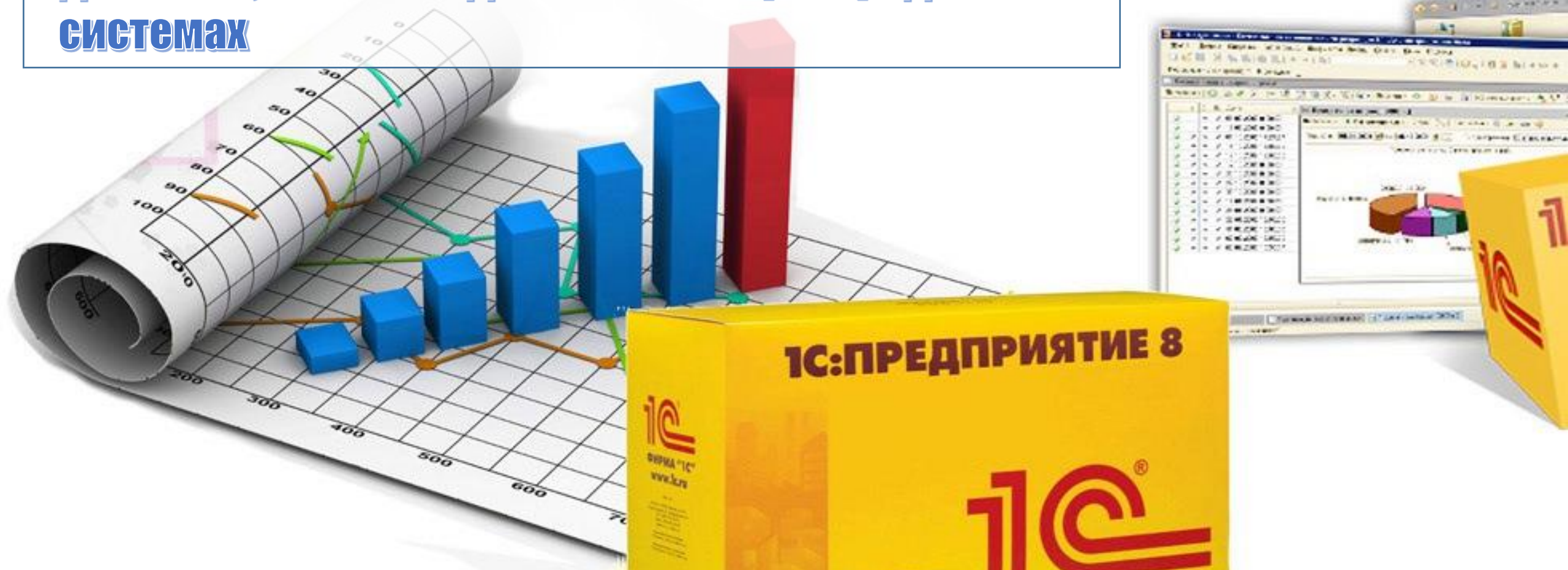
Планы обмена

При организации постоянного обмена может возникнуть ряд задач:

- ❑ 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.

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

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

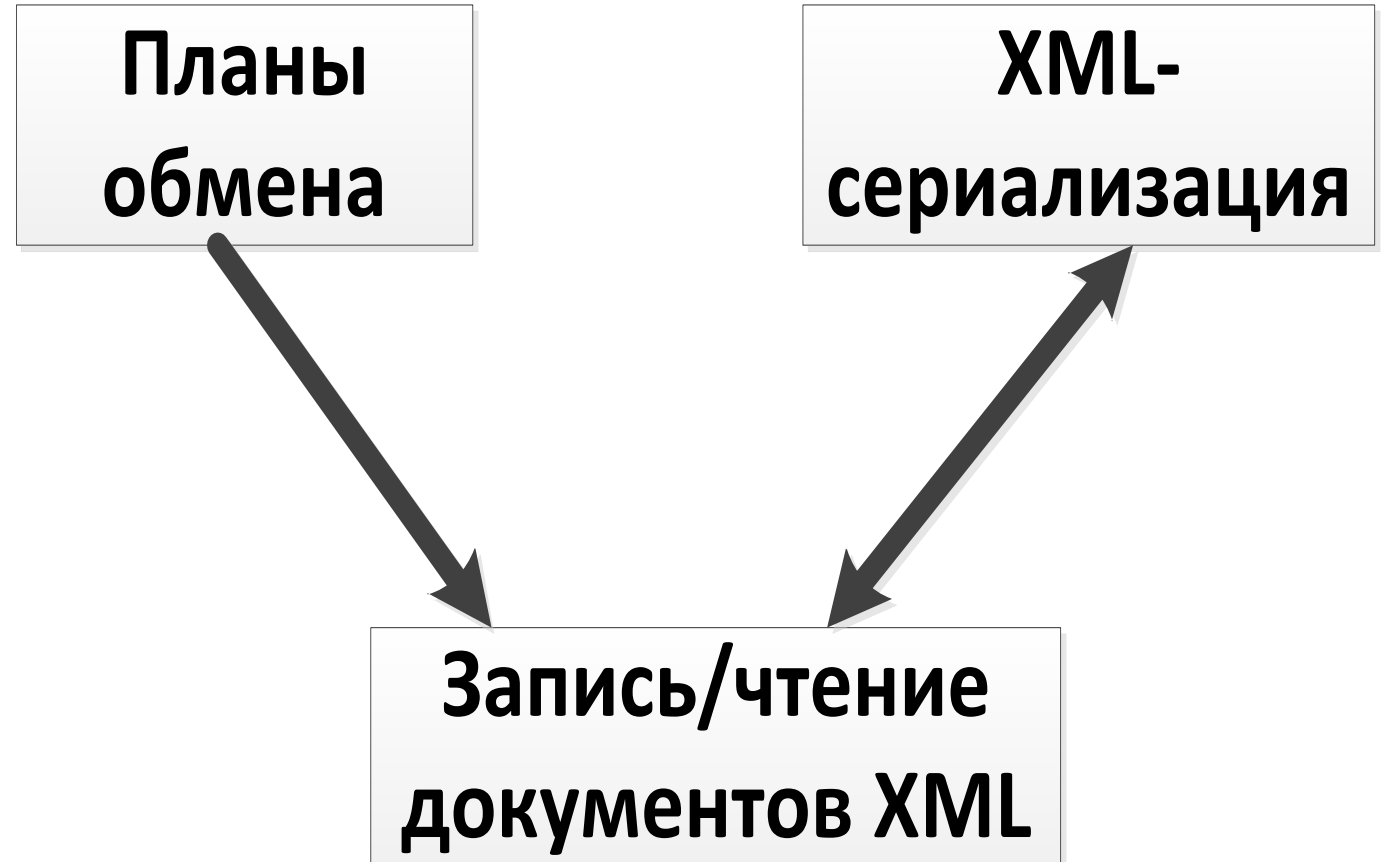


- ❑ *Универсальный механизм обмена данными.*
- ❑ *Обмен данными в распределенных системах.*

Средства платформы, используемые для построения схем обмена данным

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.



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:

- 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.

Сравнение механизмов обмена

	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.
- ❑ 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.
- ❑ 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.

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.
- ❑ Функцию формирования префикса номера вынесем в общий модуль Обмен. Она возвращает значение константы ПрефиксНумерации

```
Общий модуль Обмен: Модуль
- функция ПолучитьПрефиксНомера () Экспорт
    Возврат Константы.ПрефиксНумерации.Получить ();
- Конецфункции
```

Доработка объектов конфигурации, участвующих в обмене (продолжение)

- 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:

```
[- Процедура ПриУстановкеНовогоКода (СтандартнаяОбработка, Префикс)  
    Префикс = Обмен.ПолучитьПрефиксНомера ();  
- КонецПроцедуры
```

Доработка объектов конфигурации, участвующих в обмене (продолжение)

- Добавление плана обмена

План обмена Филиалы

Основные

Подсистемы

Функциональные опции

Данные

Формы

Поле ввода

Команды

Макеты

Ввод на основании

Права

Прочее

Имя: Филиалы

Синоним: Филиалы

Комментарий:

Распределенная информационная база

Состав

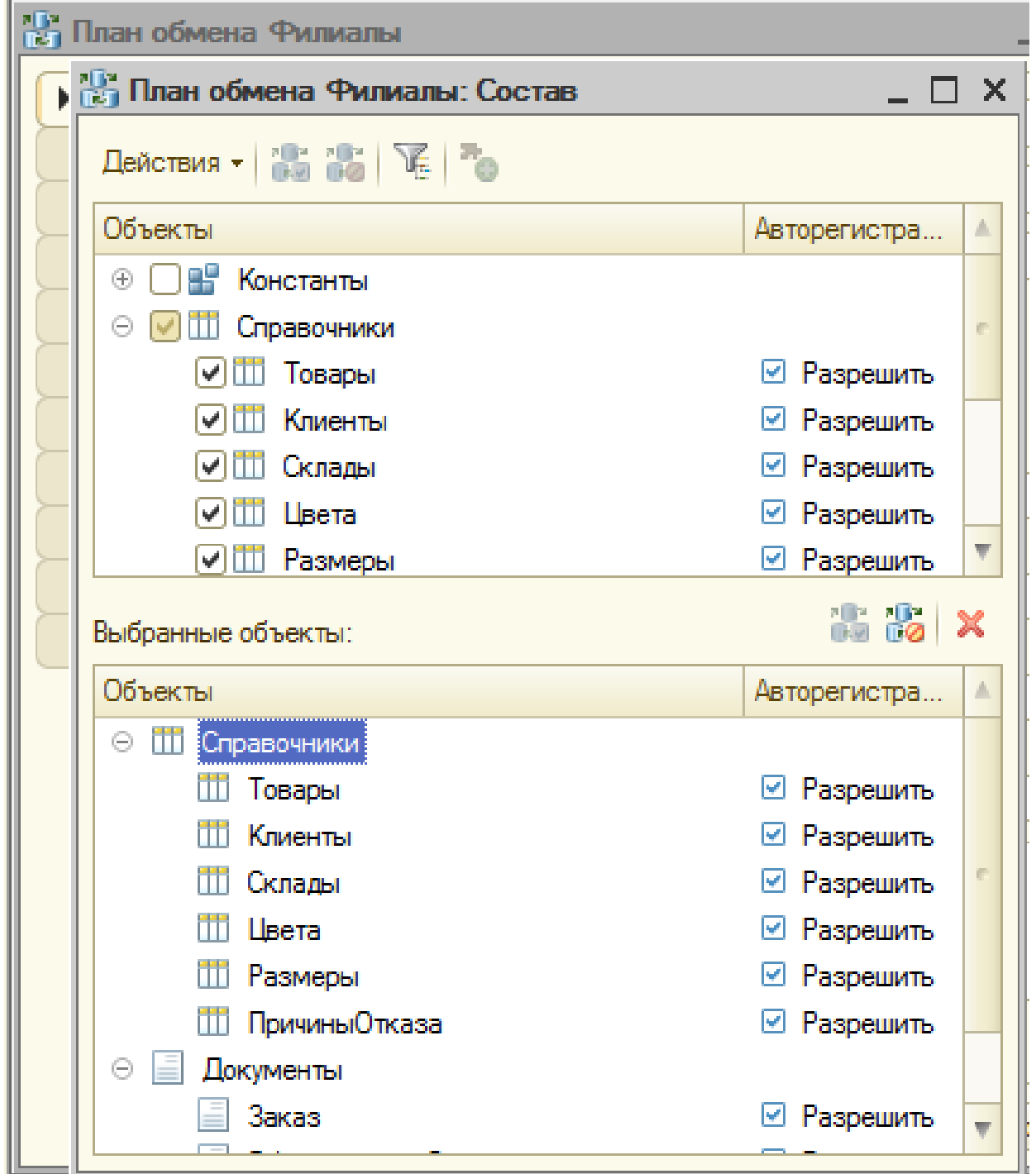
Представление объекта:

Расширенное представление объекта:

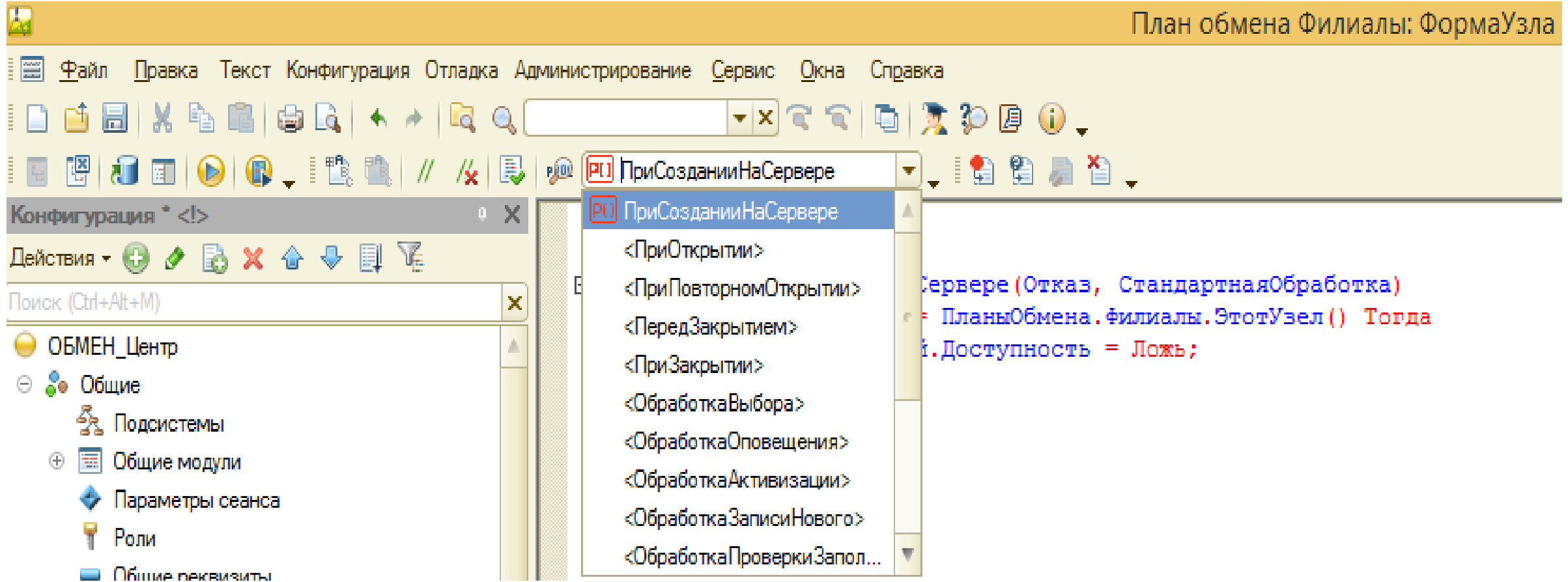
Представление списка:

Доработка объектов конфигурации, участвующих в обмене (продолжение)

- Настройка состава объектов в плане обмена



Обработчик события формы «При создании на сервере»



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


Обработчик команды на клиенте и на сервере без контекста формы

&НаСервереБезКонтекста

```
[-] Процедура ЗарегистрироватьИзмененияНаСервере (Узел)  
    // Регистрация изменений всех данных для узла.  
    ПланыОбмена.ЗарегистрироватьИзменения (Узел) ;  
[-] КонецПроцедуры
```

&НаКлиенте

```
[-] Процедура ЗарегистрироватьИзменения (Команда)  
    ЗарегистрироватьИзмененияНаСервере (Элементы.Список.ТекущаяСтрока) ;  
[-] КонецПроцедуры
```

Обработчик события ПриАктивизацииСтроки

План обмена Филиалы: ФормаСписка

1280x1024 SXGA

Форма

- Командная панель
 - ЗарегистрироватьИзменения
- СписокКомпонщикНастроекПользовательскиеНастройки
 - Список
 - Командная панель
 - Строка поиска
 - Состояние просмотра
 - Управление поиском
 - Код
 - Наименование

Элементы | Командный интерфейс

Создать | Зарегистрировать изменения | Поиск (Ctrl+F)

Поле1: Значение1 | Поле2: Значение2

Код	Наименование
-----	--------------

Свойства: Таблица

Поиск (Ctrl+Alt+I)

АвтоМаксимальнаяВысота

ВысотаВСтрокахТаблицы

АвтоМаксимальнаяВысотаВСтрокахТаблицы

ГоризонтальноеПоложениеВГруппе Авто

ВертикальноеПоложениеВГруппе Авто

РастягиватьПоГоризонтали

РастягиватьПоВертикали

ВысотаШапки

ВысотаПодвала

ВысотаЗаголовка

События:

ПриИзменении	<input type="text"/>
Выбор	<input type="text"/>
ПриАктивизацииСтроки	Список.ПриАктивизацииСтроки
ВыборЗначения	<input type="text"/>
ПриАктивизацииПоля	<input type="text"/>
ПриАктивизацииЯчейки	<input type="text"/>

Код обработчика события ПриАктивизацииСтроки

&НаСервереБезКонтекста

[-] функция ПредопределенныйУзел (Узел)

 Возврат Узел = ПланыОбмена.филиалы.ЭтотУзел();

[-] Конецфункции

&НаКлиенте

[-] Процедура СписокПриАктивизацииСтроки (Элемент)

 Если ПредопределенныйУзел (Элемент.ТекущаяСтрока) Тогда

 Элементы.ЗарегистрироватьИзменения.Доступность = Ложь;

 Иначе

 Элементы.ЗарегистрироватьИзменения.Доступность = Истина;

 КонецЕсли;

[-] КонецПроцедуры

Код обработчика команды ВыполнитьОбмен

‡НаКлиенте

```
[-] Процедура ВыполнитьОбмен (Команда)
      ВыполнитьОбменНаСервере ();
КонецПроцедуры
```

‡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 ();
          КонецЕсли;
      КонецЦикла;
КонецПроцедуры
```

Процедура записи данных обмена

```
□ 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);
КонецЦикла;

ЗаписьСообщения.ЗакончитьЗапись();
ЗаписьXML.Закрыть();

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

КонецПроцедуры

Код процедуры чтения данных обмена

□ 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
    Возрат;
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();
    Возрат;
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);
```

Код процедуры чтения данных обмена (продолжение)

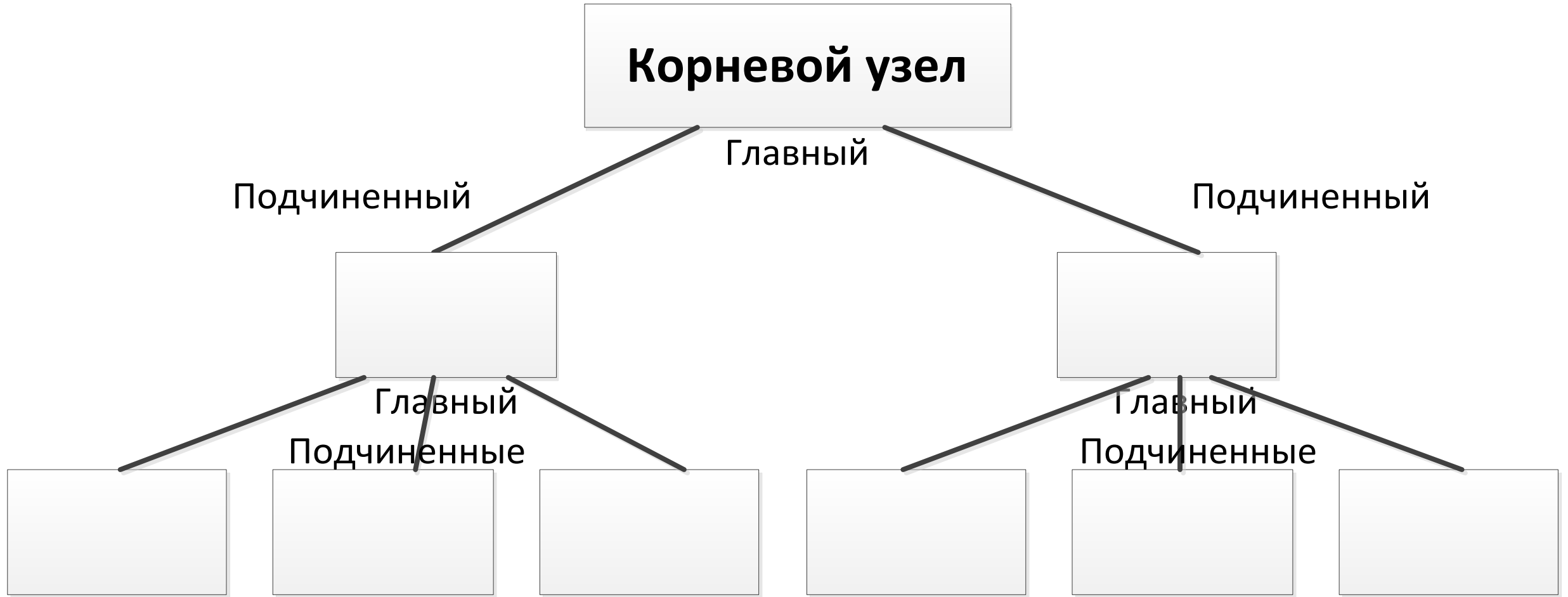
```
// Читаем данные из сообщения *** XML-сериализация.  
Пока ВозможностьЧтенияXML(ЧтениеXML) Цикл  
    // Читаем очередное значение.  
    Данные = ПрочитатьXML(ЧтениеXML);  
    // Не переносим изменение, полученное в главный из неглавного, если есть регистрация изменения.  
    Если Не ЧтениеСообщения.Отправитель.Главный 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
```

- ❑ 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;
- 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;

Механизм распределенных информационных баз



- ❑ Механизм распределённых информационных баз 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.***

- ❑ Распределенная информационная база должна иметь четко определенную **древовидную структуру**. 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

Тема 9. Механизм «1С:Аналитика»



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

Механизм «1С:Аналитика»

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

□ «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»

The screenshot displays the 1C Analytics desktop interface. At the top, there is a search bar with the text "Поиск диаграмм, дашбордов, папок..." and a "Создать" button. Below the search bar is a table listing various reports. The table has columns for "Название", "Автор", "Дата изменения", and a menu icon. A context menu is open over the menu icon of the report "Товары на складах".

Название	Автор	Дата изменения	Меню
Заказы клиентов по группам товаров	Я	13:14 19.05.2021	⋮
Реализация товаров и услуг	Администратор3	12:30 19.05.2021	⋮
План факт продаж - пример отчета	Администратор	17:17 12.05.2021	⋮
Заказы клиентов график - пример отчета	Администратор	17:17 12.05.2021	⋮
Товары на складах	Я	18:39 13.04.2021	⋮
Реализация товаров по месяцам по менеджерам	Я	20:29 05.04.2021	⋮
Приобретение реализация товаров (прибыль)	Я	22:41 23.03.2021	⋮
Реализация товаров за год по менеджерам	Я	22:06 23.03.2021	⋮
Реализация товаров и услуг.Товары - Отчет 4	Я	22:01 23.03.2021	⋮
Реализация товаров и услуг.Товары - Отчет 3	Я	21:43 23.03.2021	⋮
Заказы клиентов - Отчет 1	Я	16:27 23.03.2021	⋮
План факт продаж - Отчет 2	Я	15:57 23.03.2021	⋮
Реализация товаров за год по контрагентам	Я	15:49 23.03.2021	⋮
Реализация групп товаров за год	Я	15:14 23.03.2021	⋮
Заказы клиентов по годам	Я	23:20 09.03.2021	⋮

The context menu for "Товары на складах" includes the following options:

- Переименовать
- Удалить
- Выбрать
- Переместить в...
- Копировать в...

Особенности и преимущества механизма

- «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»

Особенности и преимущества механизма

- ❑ 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»

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.

Общая схемы работы в «1С:Аналитике» (продолжение)

4) **Детализация данных.** При необходимости можно «раскрывать» полученные данные до отдельных первичных документов «1С:Предприятия».

5) **Публикация диаграммы.**

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

6) **Создание дашборда.**

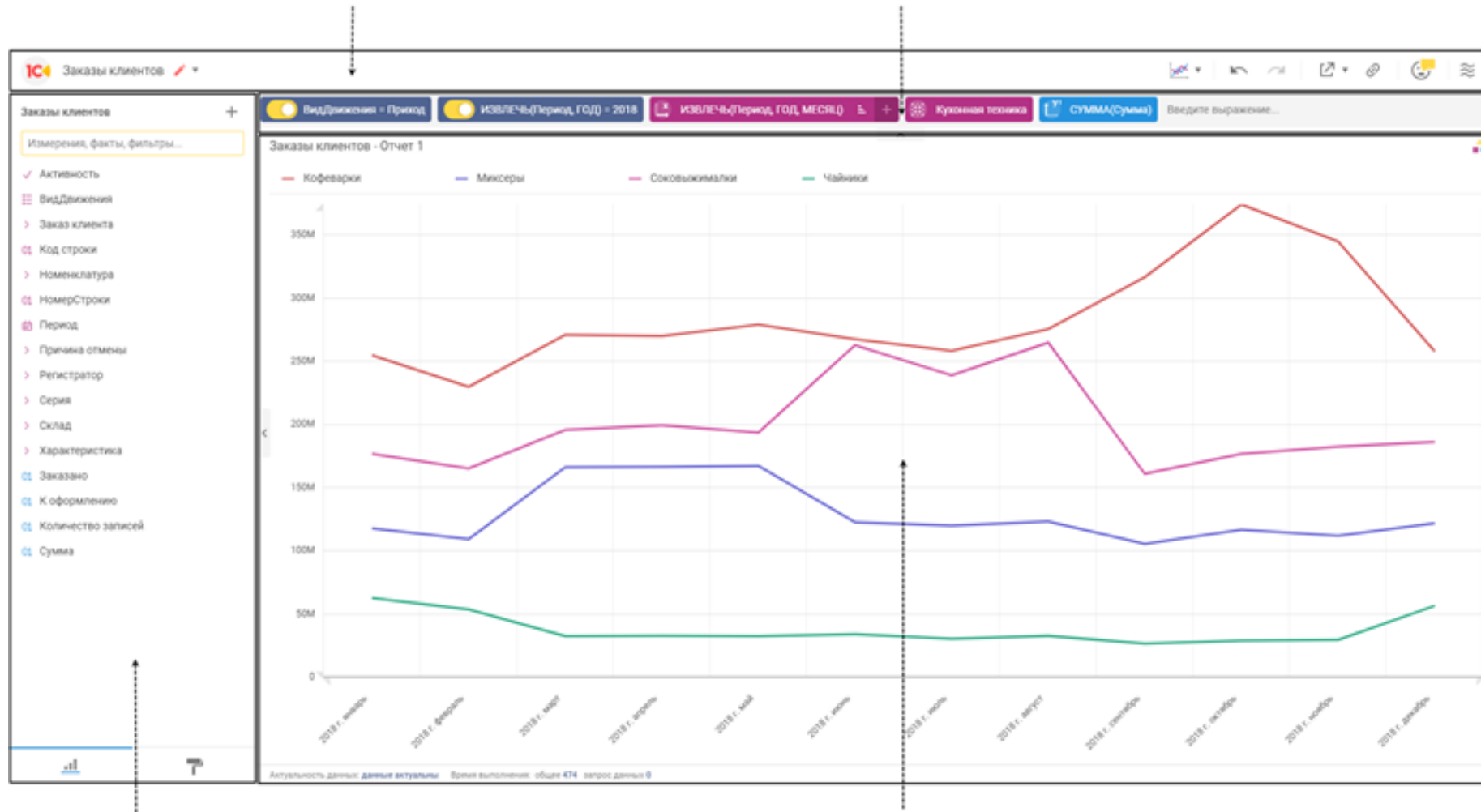
Для удобства сводного представления данных и показа нескольких связанных друг с другом диаграмм вы можете создать дашборд и вывести на него все необходимые диаграммы для комплексного представления данных по нужной тематике.

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.).

Панель источника данных и панель состава

Командная панель

Панель состава



Панель
источника данных

Рабочая область

Командная панель диаграммы

The screenshot displays the 1C software interface with three dropdown menus open, each highlighted by a red circle. The top-left menu is for document actions, the top-middle for chart types, and the top-right for user settings.

1C Товары на складах [Red Circle]

- Черновик
- Я · 16:42:58 · 13.04.2021
- ✓ Опубликовать
- 👁️ Просмотр
- ↶ Вернуться к оригиналу
- 📄 Создать копию...

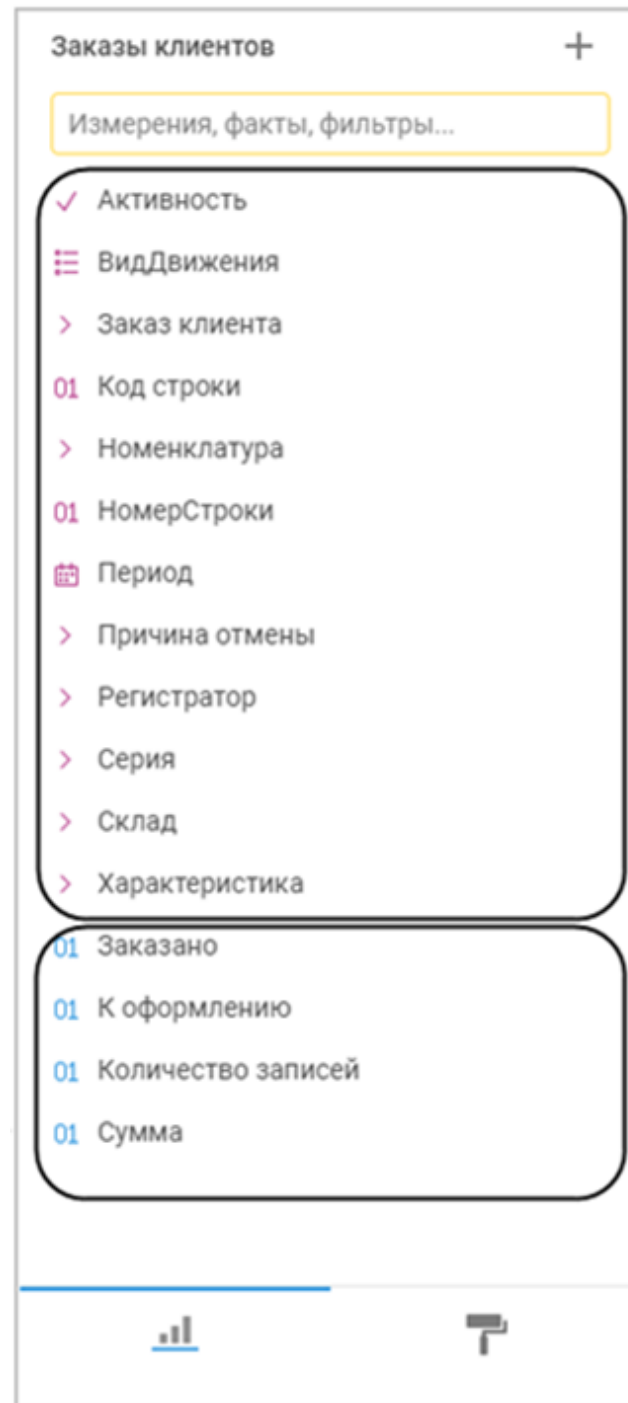
[Red Circle]

- 📊 Сводная таблица
- 📈 График
- 📊 Линии с накоплением
- 📊 Пузырьковая диаграмма
- 📊 Круговая диаграмма
- 📊 Столбчатая диаграмма
- 📊 Линейчатая диаграмма
- 📊 Вертикальный стек
- 📊 Горизонтальный стек
- 📊 Индикатор

[Red Circle]

- 🔄 Переименовать
- ↶ Отменить действие
- ↷ Повторить действие
- 📄 Экспорт >
- 🔗 Создать ссылку
- 🗣️ Оставить отзыв
- 🚀 Дата акселератор
- ✓ Служебная информация
- 🌑 Темная тема
- 📢 Центр оповещений
- 1С:Предприятие
- О программе
- ➔ Выйти Администратор

Панель источника данных



Панель состава диаграммы

Вид Движения = Приход

ИЗВЛЕЧЬ(Период, ГОД) = 2018

ИЗВЛЕЧЬ(Период, ГОД, МЕСЯЦ) = Кухонная техника

СУММА(Сумма) Введите выражение...

- || Выключить
- ▼ Сохранить фильтр
- [...] Использовать как измерение
- ✎ Редактировать
- 🗑 Удалить

По убыванию

✓ По возрастанию

Ограничить количество - 10 +

✓ Год, Месяц 2021 г. февраль

Год, Квартал 2021 г. 1 кв.

Год, Месяц, День 2021 г. февраль 26

Год 2021 г.

Квартал 1 кв.

Месяц февраль

День 26

Полная дата

✓ Ось X

Линии

📄 Сохранить выражение

▼ Создать фильтр >

[...] Использовать как факт

✎ Редактировать

🗑 Удалить

По убыванию

По возрастанию

✓ Сумма

Среднее

Максимум

Минимум

Предыдущая группировка

Нарастающий итог

Предыдущее значение

Разница

Разница, %

Доля, %

Округление

Интерполяция

Правая Ось Y

✓ Левая Ось Y

Маркеры

📄 Сохранить выражение

▼ Создать фильтр

[...] Использовать как измерение

✎ Редактировать

🗑 Удалить

□ **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. Это меню содержит как общие функции, так и функции, специфичные для роли поля.

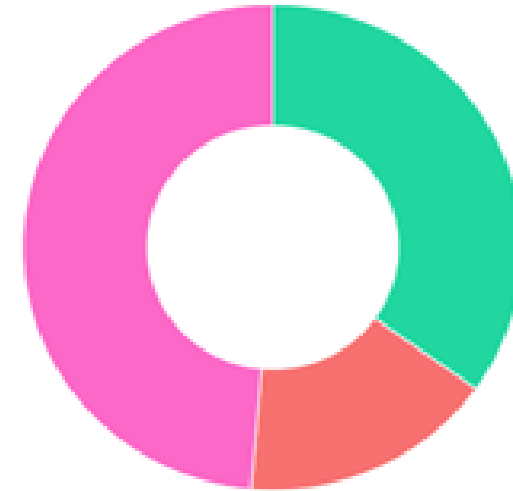
□ **Рабочая область.** В рабочей области пользователь видит результат построения диаграммы. Результат меняется «на лету», при каждом изменении состава полей диаграммы и их настроек, а также при изменении графического представления диаграммы

Пример диаграммы: Продажи товаров по менеджерам в виде круговой диаграммы

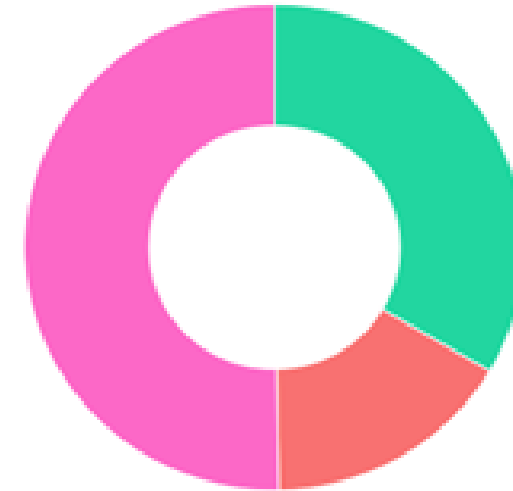
Орехов Вадим Геннадьев... Федоров Борис Михайло... Шевченко Ольга Викторо...



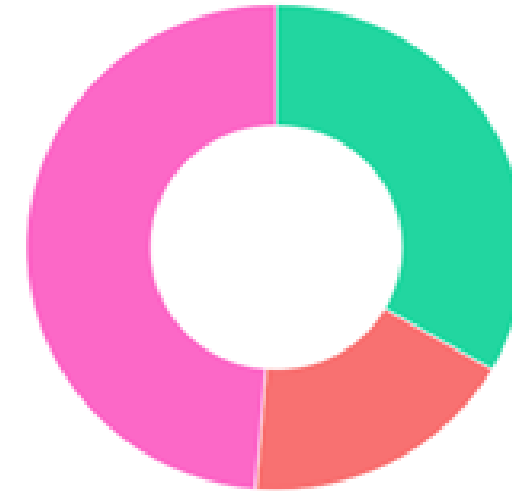
1 кв.



2 кв.

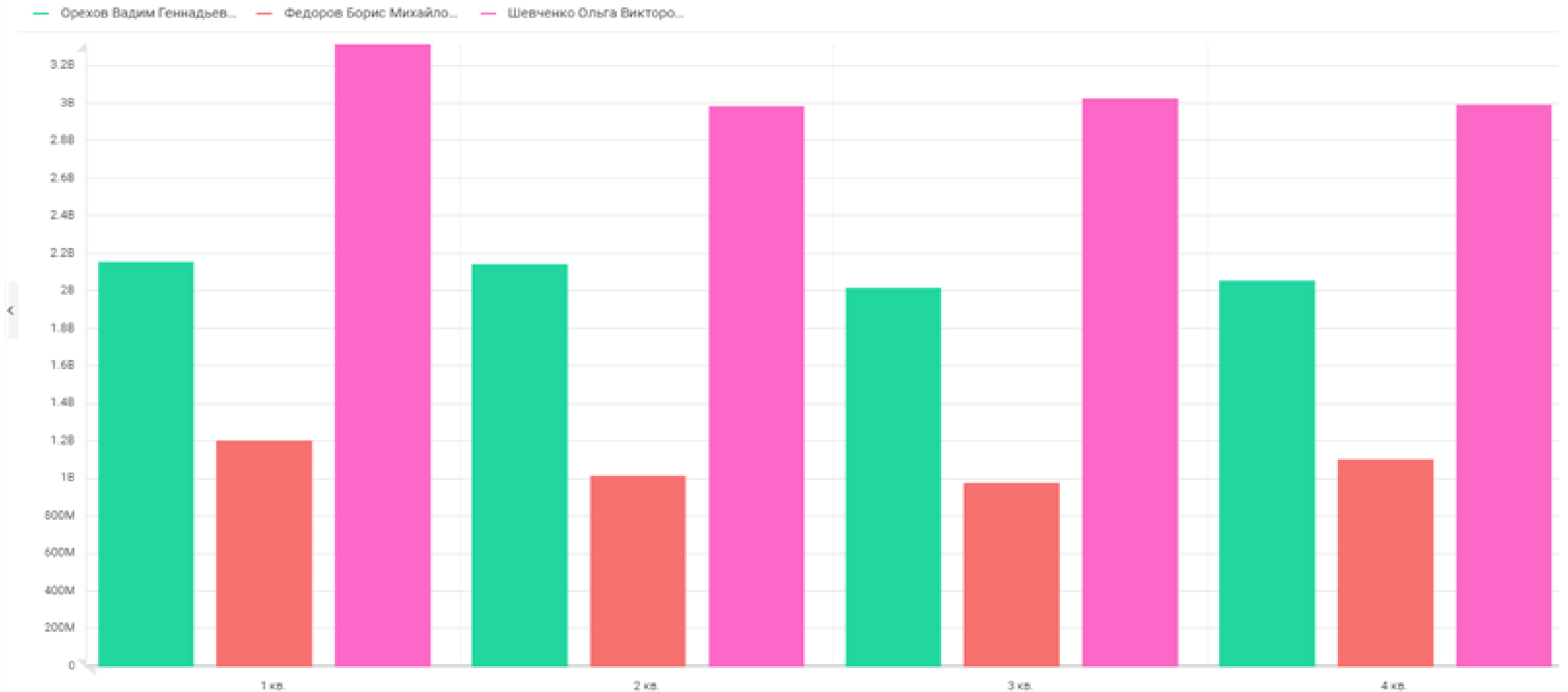


3 кв.



4 кв.

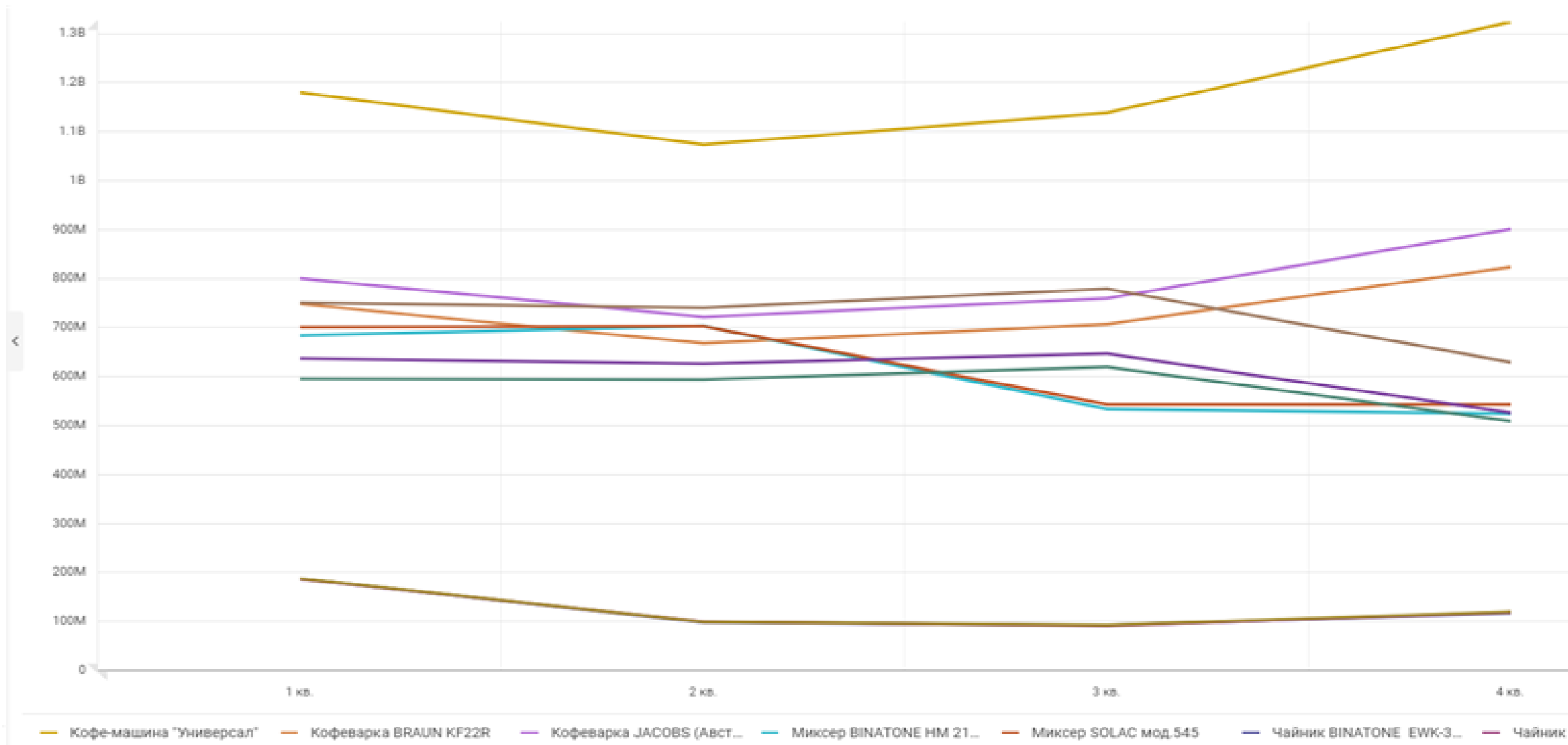
Пример диаграммы: Продажи товаров по менеджерам в виде столбчатой диаграммы



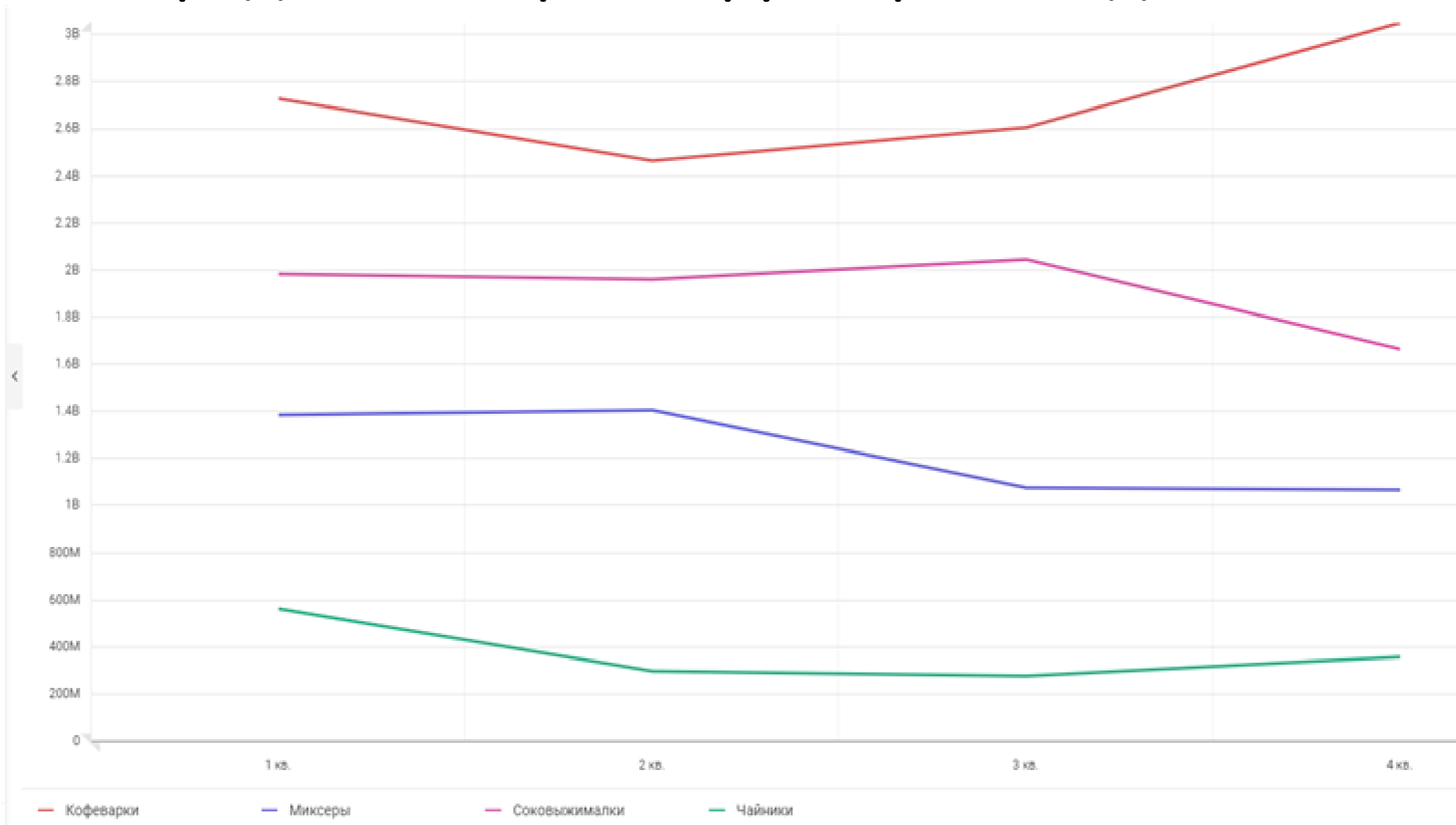
Группировка данных

- ❑ 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.

Продажи товаров без группировки данных



Продажи товаров с группировкой данных

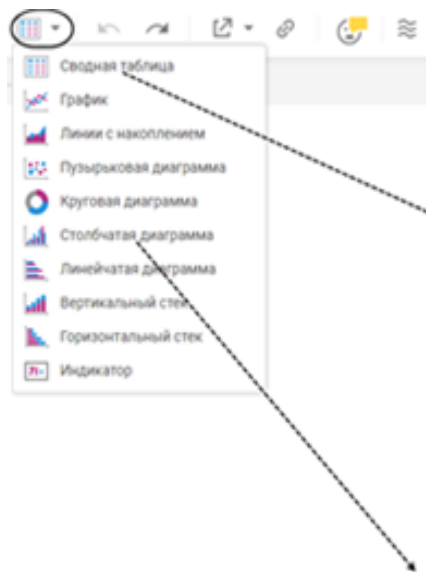


Графическое представление диаграмм

Данные могут быть представлены в виде:

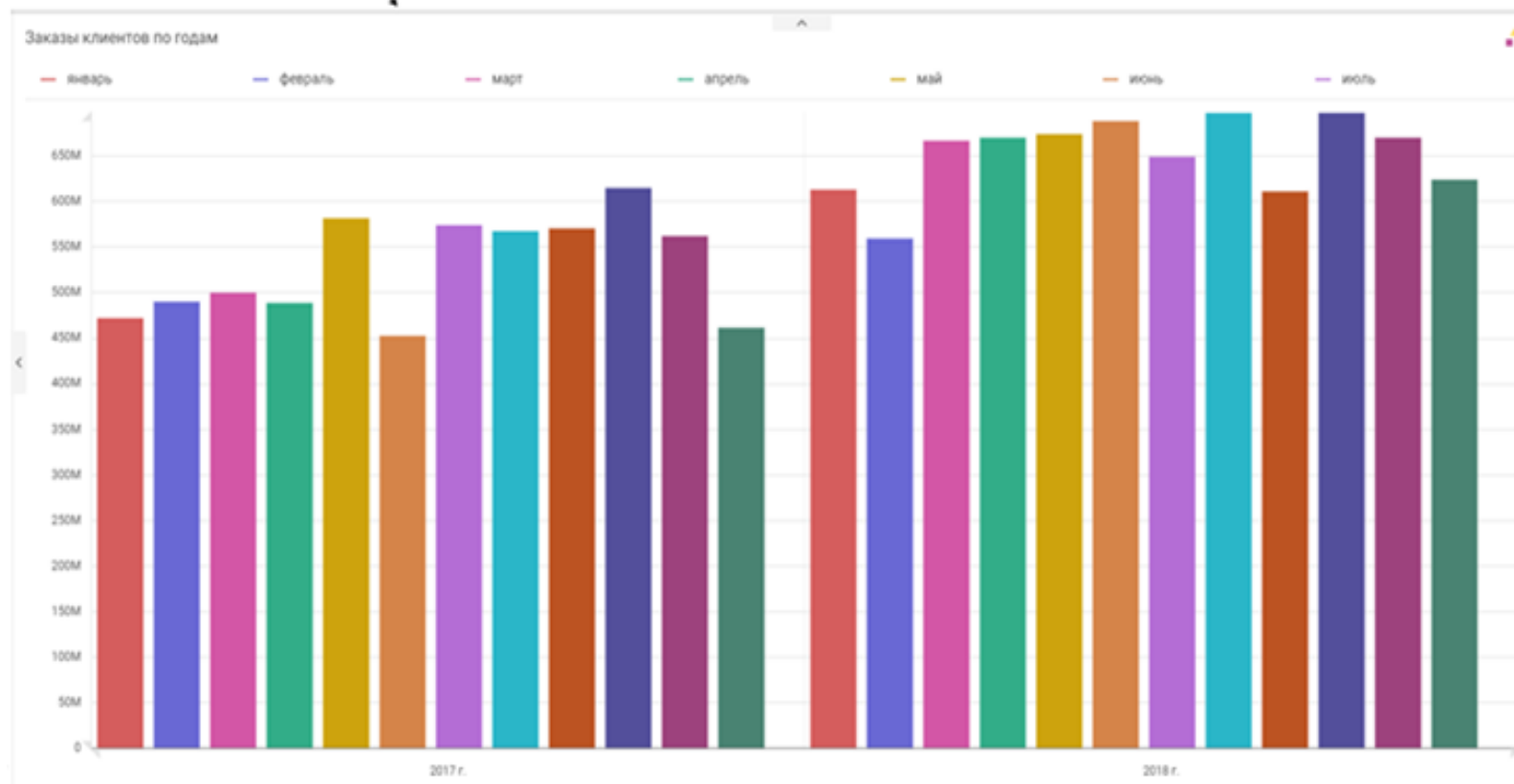
- сводной таблицы;
- графика;
- линий с накоплением;
- пузырьковой диаграммы;
- круговой диаграммы;
- столбчатой диаграммы;
- линейчатой диаграммы;
- вертикального стека;
- горизонтального стека;
- индикатора.

Выбор графического представления диаграммы



Заказы клиентов по годам

Год	2017 г.		2018 г.	
	ИЗВЛЕЧЬ(Период, ...	Сумма заказов	Сумма заказов	Сумма заказов
январь	472 274 602,8	613 316 982,6		
февраль	490 409 217,9	559 653 425,1		
март	500 188 644,9	667 040 833,2		
апрель	488 921 608,3	670 385 154,6		
май	581 815 182,5	674 192 948,1		
июнь	452 966 612,7	688 513 513,8		
июль	574 229 302,3	649 359 494,7		
август	567 823 620,0	697 992 159,9		
сентябрь	570 660 638,1	611 444 387,4		
октябрь	615 140 866,2	698 120 308,2		
ноябрь	562 522 327,5	670 402 895,4		
декабрь	461 873 907,6	624 221 201,1		



Дизайнер диаграммы

Формат диаграммы

Свойство

ИТОГИ

Отображать итоги в конце



ИТОГИ

Отображать итоги в конце



Вид: Движения - Приход | В(ИЗВЛЕЧЬ(Период, ГОД), 2017, 2018) | Год | Месяц | Кол-во заказов | Сумма заказов

Заказы клиентов по годам

Год	Итого		2017 г.		2018 г.	
	Кол-во заказов	Сумма заказов	Кол-во заказов	Сумма заказов	Кол-во заказов	Сумма заказов
Итого	1 811 972	14 189 475 634,9	815 046	6 364 832 330,8	996 926	7 824 643 304,1
январь	164 529	1 109 207 385,4	71 407	495 890 402,8	93 122	613 316 982,6
февраль	155 395	1 050 062 643,0	72 143	490 409 217,9	83 252	559 653 425,1
март	146 764	1 169 619 478,1	64 385	502 578 644,9	82 379	667 040 833,2
апрель	148 940	1 159 306 762,9	66 085	488 921 608,3	82 855	670 385 154,6
май	154 658	1 256 008 130,6	71 675	581 815 182,5	82 983	674 192 948,1
июнь	141 740	1 141 480 126,5	56 320	452 966 612,7	85 420	688 513 513,8
июль	150 026	1 223 588 797,0	70 529	574 229 302,3	79 497	649 359 494,7
август	154 980	1 265 815 779,9	69 589	567 823 620,0	85 391	697 992 159,9
сентябрь	138 062	1 182 105 025,5	66 498	570 660 638,1	71 564	611 444 387,4
октябрь	152 962	1 313 261 174,4	72 389	615 140 866,2	80 573	698 120 308,2
ноябрь	144 687	1 232 925 222,9	65 794	562 522 327,5	78 893	670 402 895,4
декабрь	159 229	1 086 095 108,7	68 232	461 873 907,6	90 997	624 221 201,1

Год	2017 г.		2018 г.		Итого	
	Кол-во заказов	Сумма заказов	Кол-во заказов	Сумма заказов	Кол-во заказов	Сумма заказов
январь	71 407	495 890 402,8	93 122	613 316 982,6	164 529	1 109 207 385,4
февраль	72 143	490 409 217,9	83 252	559 653 425,1	155 395	1 050 062 643,0
март	64 385	502 578 644,9	82 379	667 040 833,2	146 764	1 169 619 478,1
апрель	66 085	488 921 608,3	82 855	670 385 154,6	148 940	1 159 306 762,9
май	71 675	581 815 182,5	82 983	674 192 948,1	154 658	1 256 008 130,6
июнь	56 320	452 966 612,7	85 420	688 513 513,8	141 740	1 141 480 126,5
июль	70 529	574 229 302,3	79 497	649 359 494,7	150 026	1 223 588 797,0
август	69 589	567 823 620,0	85 391	697 992 159,9	154 980	1 265 815 779,9
сентябрь	66 498	570 660 638,1	71 564	611 444 387,4	138 062	1 182 105 025,5
октябрь	72 389	615 140 866,2	80 573	698 120 308,2	152 962	1 313 261 174,4
ноябрь	65 794	562 522 327,5	78 893	670 402 895,4	144 687	1 232 925 222,9
декабрь	68 232	461 873 907,6	90 997	624 221 201,1	159 229	1 086 095 108,7
Итого	815 046	6 364 832 330,8	996 926	7 824 643 304,1	1 811 972	14 189 475 634,9

Настройка и выбор цветовой схемы диаграммы

ЛЕГЕНДА

Отображать по умолчанию

Реализация товаров за год по контрагентам

- Сириус
- Альфа
- Альтаир
- Сатурн

НАСТРОЙКИ ЦВЕТА

Схема

- Моя цветовая с...
- Моя цветовая схема
- Схема не выбрана
- Стандартная цветовая схема
- Настройка схем...

Цветовые схемы

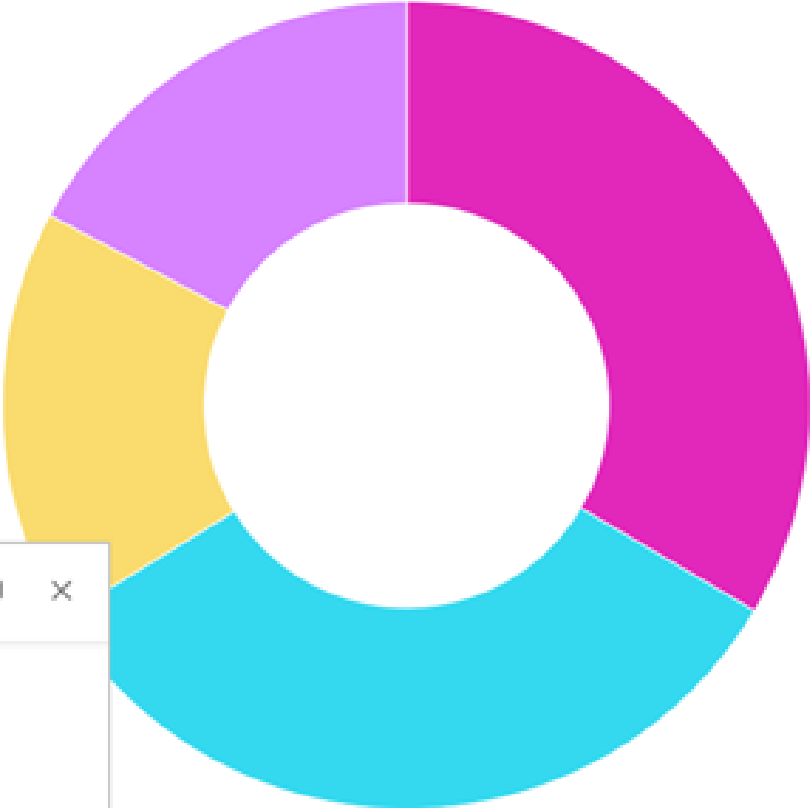
Темная тема Готово

Введите название схемы +

Моя цветовая схема

Светлая тема

× × × × × +



Работа с дашбордами

- ❑ **Дашборд** можно охарактеризовать как некий фотоколлаж, состоящий из набора нескольких виджетов, предназначенный для комплексного представления данных по нужной тематике.
- ❑ Как правило, виджеты содержат отдельные тематически связанные друг с другом диаграммы. Они могут менять свое местоположение и размер.
- ❑ 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 этот виджет.

Дашборд с данными о заказах и о реализации товаров за год

Сумма заказов клиентов по месяцам 2018 г

624 221 201,1
↓ -6.89%



Сумма реализации товаров по месяцам 2018 г

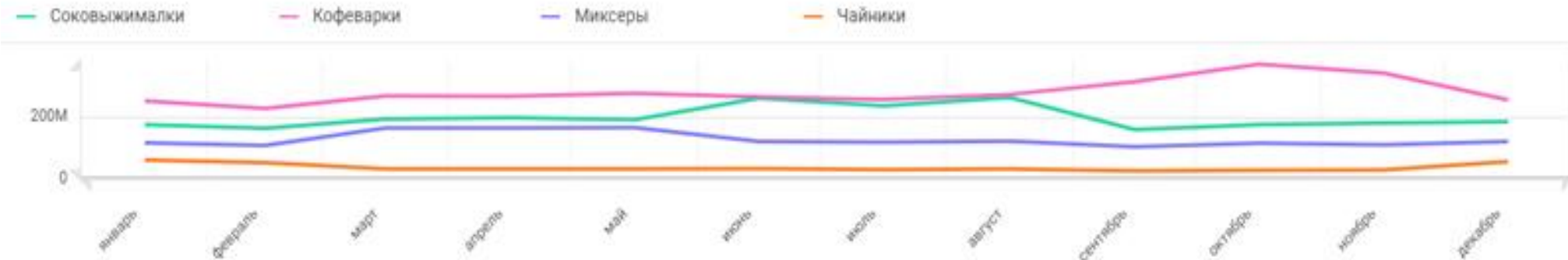
1 868 273 328,6
↓ -10.65%



План факт продаж



Заказы клиентов по группам товаров



Условное оформление элементов диаграммы

КРАСИТЬ(СУММА(ФактКоличество), ЕСЛИ(Условное_Оформление_План_факт) 2018 год Месяц Факт кол-во План кол-во Введите выражение...

План факт продаж - условное оформление

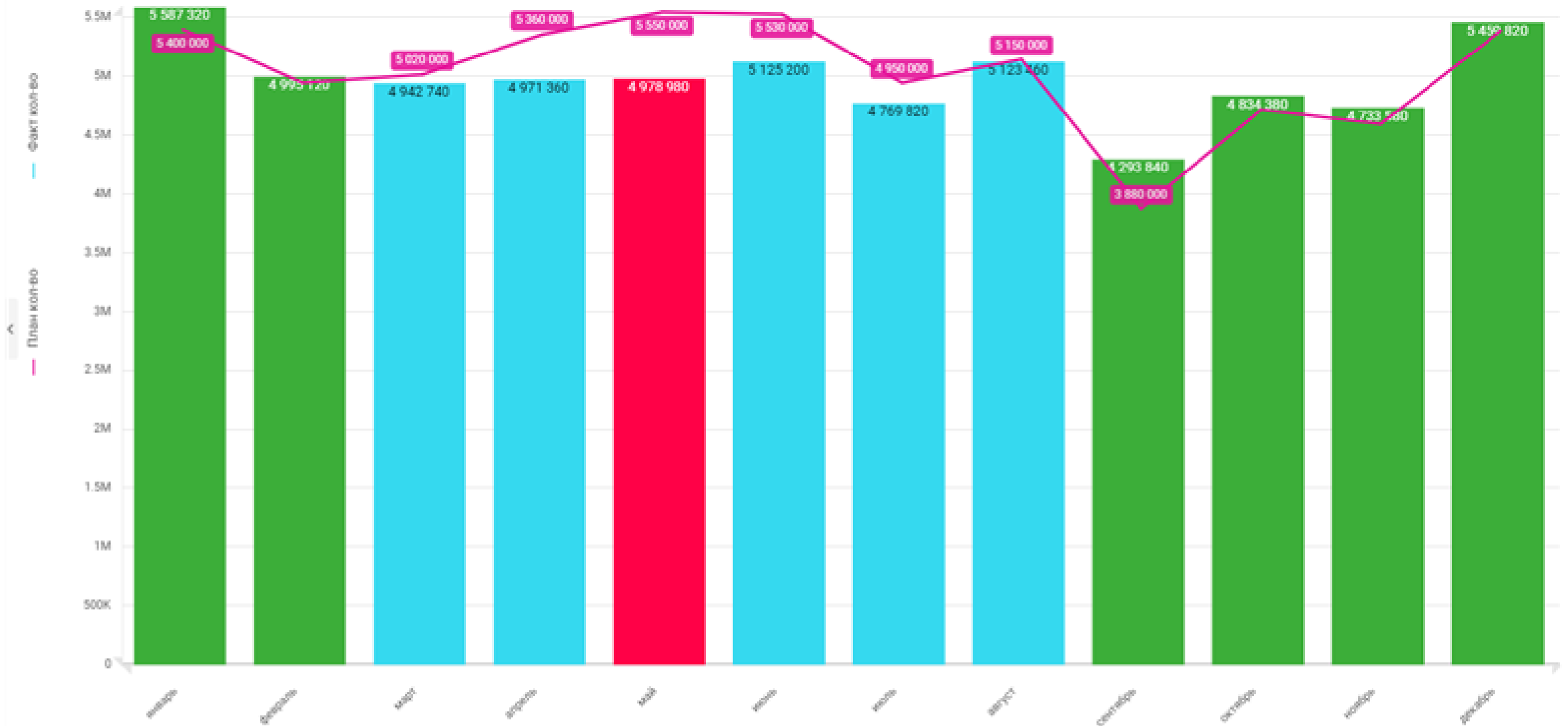


Схема развертывания «1С:Аналитика»

