

Оглавление

Вопрос №1	Прикладные системы обработки данных.....	2
Вопрос №2	Прикладная система обработки данных 1С:Предприятие	3
Вопрос №3	Структура технологической платформы 1С:Предприятие	4
Вопрос №4	Виды клиентских частей платформы 1С:Предприятие	5
Вопрос №5	Сравнение платформы 1С:Предприятие с классическими СУБД.	7
Вопрос №6	Механизм запросов и работа с БД в 1С:Предприятие	8
Вопрос № 7	Метаданные конфигурации	9
Вопрос №8	Справочники	10
Вопрос №9	Документы	12
Вопрос №10	Проведение документа	13
Вопрос №11	Формы документа	14
Вопрос №12	Многомерные хранилища данных	15
Вопрос №13	Регистры накопления	17
Вопрос №14	Виды регистров накопления.....	18
Вопрос №15	Правильное проектирование регистров накопления	19
Вопрос №16	Индексация измерений регистра накопления.....	20
Вопрос №17	Функциональные возможности регистра накопления	21
Вопрос №18	Регистры сведений.....	22
Вопрос №19	Константы	24
Вопрос № 20	Перечисления	25
Вопрос № 21	Механизм характеристик.....	26
Вопрос № 22	Механизмы запросов 1С.....	28
Вопрос № 23	Ссылочные типы данных	29
Вопрос № 24	Хранение данных ссылочных типов в базе данных.....	30
Вопрос № 25	Хранение данных нессылочных типов.....	32
Вопрос № 26	Источники данных для запросов.....	33
Вопрос № 27	Синтаксис текста запросов.....	34
Вопрос №28	Раздел ИЗ.....	Error! Bookmark not defined.
Вопрос № 29	Раздел ВЫБРАТЬ	37
Вопрос № 30	Раздел ГДЕ	38
Вопрос № 31	Раздел СГРУППИРОВАТЬ ПО	39
Вопрос № 32	Раздел ИМЕЮЩИЕ	40
Вопрос № 33	Раздел УПОРЯДОЧИТЬ ПО	41
Вопрос № 34	Раздел ИТОГИ ПО.....	42
Вопрос № 35	Раздел ОБЪЕДИНИТЬ ВСЕ.....	43
Вопрос №36	Табличный документ и принципы работы с ним.....	44
Вопрос № 37	Component Object Model (COM)	45
Вопрос №38	Отчеты	46
Вопрос №39	Система компоновки данных	47
Вопрос № 40	Схема компоновки данных.....	49
Вопрос №41 Задача		50
Вопрос №42 Задача		51
Вопрос №43 Задача		52

Вопрос №1 Прикладные системы обработки данных

Прикладная система обработки данных предназначена для выполнения узких задач и рассчитана на непосредственное взаимодействие с пользователем. В большинстве операционных систем прикладные программы не могут обращаться к ресурсам компьютера напрямую, а взаимодействуют с оборудованием и другими программами посредством операционной системы.

Также на простом языке — вспомогательные программы.

К прикладному программному обеспечению относятся компьютерные программы, написанные для пользователей или самими пользователями для задания компьютеру конкретной работы.

Программы обработки заказов или создания списков рассылки — пример прикладного программного обеспечения.

Программисты, которые пишут прикладное программное обеспечение, называют прикладными программистами.

ПСОД классифицируется по типу:

- ПО общего назначения (текстовые и графические редакторы, веб-браузеры, СУБД)
- ПО мультимедийное (видео и музыкальные плееры)
- ПО профессиональное (САПР, CRM, АРМ)

ПСОД классифицируются по сфере применения:

- ПО предприятий и организаций
- ПО, обеспечивающее доступ пользователей к ресурсам компьютера
- ПО имитационное
- ПО для проектирования и конструирования
- ПО веб-среды
- ПО управления транзакциями
- ПО рабочего места

Вопрос №2 Прикладная система обработки данных

1С:Предприятие

Абстрактно структуру среды программирования 1С Предприятие можно представить в следующем виде:



Технологическая платформа — это программная оболочка, обеспечивающая на базовом уровне выполнение вычислений, взаимодействие с источниками данных, клиентский интерфейс. Можно привести образную аналогию с музыкальным проигрывателем, который может выполнять различные аудио - файлы . Как правило, с платформой имеют дело разработчики, а не конечные пользователи.

Конфигурации или прикладные решения — это то, с чем по сути и работают конечные пользователи. Предоставляют клиентский уровень, а так же уровень обработки данных. Как правило каждое прикладное решение связано со своей информационной базой данных. Весь оперативный, налоговый, бухгалтерский учет ведется с помощью тех или иных прикладных решений. А платформа обеспечивает выполнение клиентского уровня, вычислений, взаимодействие с источниками данных. Если воспользоваться все той же аналогией с проигрывателем, то конфигурации сродни аудио-файлам. Одновременно платформа может «проигрывать» несколько конфигураций.

Внедрение - есть очень ответственный и трудоемкий процесс обучения пользователей работе с функционалом прикладного решения и первоначального ввода данных. Т.к. большинство текущих данных есть консолидации данных, накопленных в прошлом, то от первоначального ввода решающим образом зависит возможность получения адекватных данных в текущем моменте.

Вопрос №3 Структура технологической платформы

1С:Предприятие

Существует четкое разделение на платформу и прикладное решение.

Платформа представляет собой framework, в котором разрабатываются, администрируются и исполняются прикладные решения.

Прикладное решение выступает в качестве отдельного программного продукта, опирающегося на технологическую платформу.

Как и любой framework, платформа предоставляет для разработки прикладных решений суперклассы, служащие основой для создания метаданных прикладного решения. Т.е. на основе базовых объектов платформы разработчик создает типовые объекты своей конфигурации, от которых будут порождаться экземпляры пользователей.

Платформа предоставляет достаточный функционал для построения прикладных решений:

- Клиентские части: толстый, тонкий, веб-клиент
- СУБД и внутренний формат БД
- Язык запросов
- Система компоновки данных
- Система метаданных
- Построители отчетов
- COM –соединение
- XML-сериализация
- Веб-сервисы
- Система прав доступа

Вопрос №4

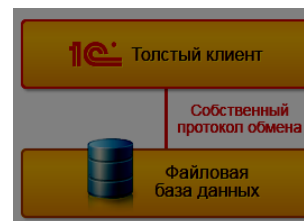
Виды клиентских частей платформы 1С:Предприятие

Толстый клиент — это компонент платформы, позволяющий реализовывать полные возможности платформы в плане исполнения прикладного кода, но не поддерживает работу с информационными базами через интернет, требует предварительной установки на компьютер пользователя и имеет довольно внушительный объем дистрибутива.

Клиент — сервер

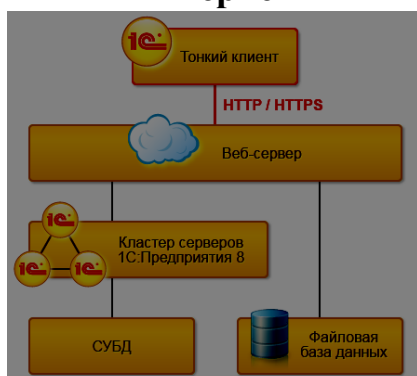


Файловая организация



Тонкий клиент - это одно из клиентских приложений системы 1С:Предприятие 8. Может работать с информационными базами через интернет. Он также требует предварительной установки на компьютер пользователя, но имеет значительно меньший размер дистрибутива, чем толстый клиент.

Интернет



Клиент-сервер



Файловая организация



При работе тонкого клиента в файловом варианте работы на компьютере, где запущен сам тонкий клиент, организуется специализированная среда, решающая такие задачи как загрузка необходимых для работы системы серверных компонентов, загрузка прикладной конфигурации, другие действия,

необходимые для организации нормальной работы системы с информационной базой.

Веб-клиент - это клиентское приложение системы 1С:Предприятие 8. Его не нужно предварительно устанавливать на компьютер пользователя. И выполняется он в среде браузера (Windows Internet Explorer, Mozilla Firefox, Google Chrome или Safari). При работе веб-клиента клиентские модули, разработанные в конфигурации, компилируются автоматически из встроенного языка 1С:Предприятия 8 и непосредственно исполняются на стороне веб-клиента.



Таким образом, независимо от клиентского приложения (толстый, тонкий, веб-клиент), вся разработка прикладного решения ведется полностью в конфигураторе 1С:Предприятия, серверный и клиентский код пишется на встроенном языке 1С:Предприятия.

Вопрос №5 Сравнение платформы 1С:Предприятие с классическими СУБД.

Объектная модель работы с данными в 1С:Предприятии имеет ряд особенностей, отличающих ее от классических моделей СУБД, и заключается в том, что разработчик не обращается к базе данных напрямую. Непосредственно он работает с платформой 1С:Предприятия. При этом он может описывать структуры данных в конфигураторе, манипулировать данными с помощью объектов встроенного языка, составлять запросы к данным, используя язык запросов. Платформа исполняет запросы, транслируя их в соответствующие команды SQL. Это могут быть команды системы управления базами данных, в случае клиент-серверного варианта работы, или команды собственного движка базы данных для файлового варианта.



Вопрос №6 Механизм запросов и работа с БД в 1С:Предприятие

Механизм запросов- это один из способов доступа к данным, которые поддерживает платформа. Используя этот механизм, разработчик может читать и обрабатывать данные, хранящиеся в информационной базе; изменение данных с помощью запросов невозможно. Это объясняется тем, что запросы специально предназначены для быстрого получения и обработки некоторой выборки из больших массивов данных, которые могут храниться в базе данных.

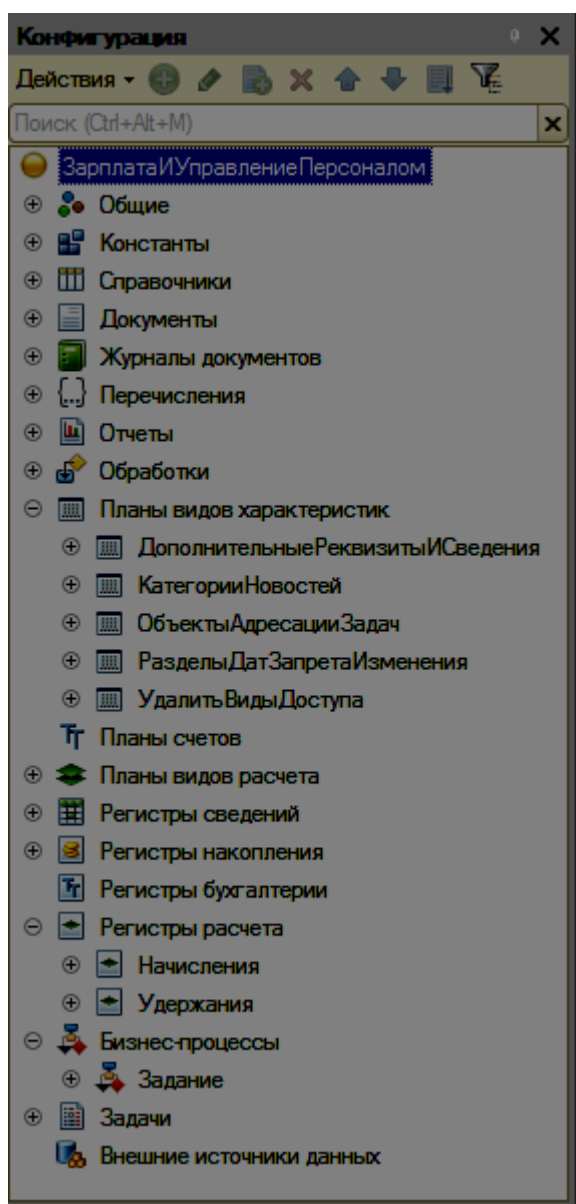
Платформа снабжена собственным языком запросов к БД. Этот язык основан на SQL, но при этом содержит значительное количество расширений таких, как обращение к табличным частям документов и элементов справочников, автоматическое упорядочивание, многоуровневое форматирование итогов, поддержка виртуальных таблиц, временных таблиц, пакетные запросы, конструктор запросов, консоль запросов

Работа с базами данных таких СУБД, как MS SQL Server, PostgreSQL, IBM DB2, ORACLE DATABASE. А так же поддерживается ядро собственной внутренней СУБД, которая используется при файловой организации

Вопрос № 7

Метаданные конфигурации

Прикладные объекты, еще говорят «объекты конфигурации» или «объекты метаданных» - это проблемно-ориентированные объекты, поддерживаемые на уровне технологической платформы. «Проблемно-ориентированность» означает направленность на оптимальную структуру и поведение объекта для конкретных задач, чаще всего в сфере экономики. Эти объекты имеют исходно заданные или типовые свойства и методы. Задача разработчика состоит в том, чтобы собрать из этих объектов, необходимую структуру прикладного решения и задать алгоритмы функционирования этих объектов, отличающиеся от их типового поведения. Разработчик может оперировать



такими прикладными объектами как справочники, документы, регистры сведений, регистры накопления, планы счетов, планы видов характеристик и т.д. На рисунке приведено дерево метаданных или дерево конфигурации, ветви которого соответствуют абстрактным классам метаданных, от которых порождаются метаданные конкретной конфигурации. Так План видов характеристик «КатегорииНовостей» - это объект метаданных или прикладной объект, относящийся только к конкретной конфигурации «ЗарплатаИУправлениеПерсоналом», а Планы видов характеристик будет встречаться во всех конфигурациях.

На основании описания разработчика платформа сама создаст в базе данных соответствующие информационные структуры, и определенным образом будет работать с данными, хранящимися в этих структурах. Разработчику нет необходимости заботиться о том, в каких таблицах, например, должны размещаться данные, каким образом они будут модифицироваться или представляться пользователю. Все эти

действия платформа будет выполнять автоматически, исходя из типового поведения используемых объектов.

Справочники - это объекты метаданных, позволяющие хранить в информационной базе данные, имеющие одинаковую структуру. Эти данные, как правило, имеют постоянный или полу постоянный характер и используются для автоматической подстановки в экранные формы и прочие объекты. Это может быть, например, список сотрудников, перечень товаров, список поставщиков или покупателей.

Структура справочника

Справочник состоит из набора информационных единиц или элементов, имеющих одинаковую структуру. Каждый элемент справочника обязательно характеризуется **кодом** и **наименованием**. Система поддерживает режим автоматической нумерации элементов. А так же позволяет осуществлять контроль уникальности кодов справочника.

Каждый элемент справочника может содержать некоторую дополнительную информацию, которая подробно описывает этот элемент. Эта информация хранится в реквизитах справочника. Например, для товара это может быть информация об артикуле, упаковке и т.д.

Каждый элемент справочника может содержать некоторый набор информации, которая одинакова по своей структуре, но различна по количеству, для разных элементов справочника. Например, это может быть контактная информация или информация о составе семьи, образовании. Для таких данных служат табличные части справочника.

Располагаться элементы справочника могут как в иерархии, так и без нее. При иерархическом расположении в справочнике создаются группы, в которых располагаются элементы. Так же в группы могут содержать другие группы, создавая тем самым многоуровневую иерархическую структуру. В справочниках возможен и другой вид иерархии, при котором элементы справочника будут относиться не к группам, а к другим элементам этого же справочника.

Элементы одного справочника могут быть подчинены элементам или группам другого справочника. Например, справочник **Кассы** может быть подчинен справочнику **Организации**. Тогда при оформлении кассовых документов для некоторой организации можно будет выбрать кассу не среди всех имеющихся в программе касс, а среди касс, существующих только в этой организации.

В справочнике могут присутствовать так же predetermined элементы, которые задаются разработчиком и не могут быть удалены пользователем.

Формы справочника

Для просмотра данных справочника, существует несколько форм представления справочника. Система может автоматически генерировать все нужные формы справочника. Разработчик имеет возможность создать собственные формы, которые система будет использовать вместо форм по умолчанию.

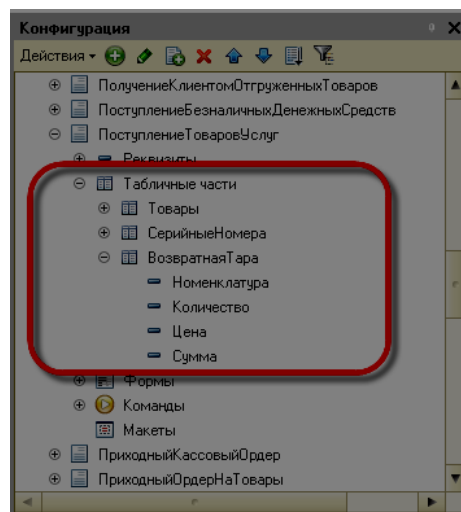
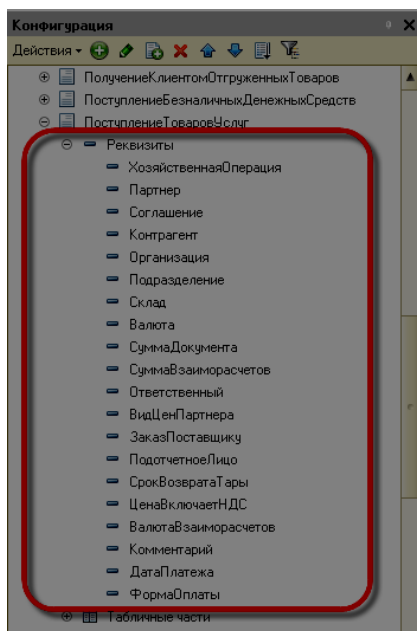
Форма элемента справочника используется для просмотра и редактирования данных элемента справочника. Форма списка отражает множество элементов справочника для просмотра. Форма выбора используется для выбора и подстановки элемента справочника в другие объекты. Форма групп используется для создания и редактирования группы справочника.

Вопрос №9

Документы

Документы - это прикладные объекты конфигурации, так же, как и справочники, являющиеся проблемно-ориентированными объектами, т.е. предназначены для оптимального решения узкого круга задач. В частности, они позволяют собирать и хранить в базе данных информацию о совершенных хозяйственных операциях. Это могут быть, например, приходные накладные, приказы о приеме на работу, счета, платежные поручения и т.д.

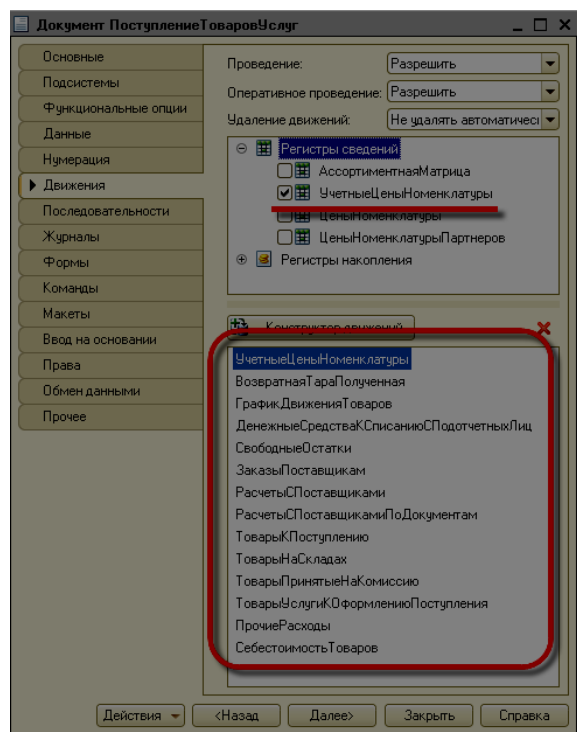
Документ характеризуется **номером, датой и временем**. Система поддерживает режим, при котором она самостоятельно может генерировать номер для нового документа. Кроме этого система позволяет осуществлять контроль уникальности номеров документов. Дата и время документа позволяют установить строгую временную последовательность совершения операций, в результате чего документы могут отличаться друг от друга не только номером, но и своим положением на временной оси. Как и справочники, документы имеют реквизиты и табличные части. В реквизитах документа хранится дополнительная информация, описывающая документ, например для документа «Поступление товаров на склад» в реквизитах может храниться информация о складе, материально ответственном лице, сопровождающем товар документе и т.д. А вот информация о перечне товаров, цене за единицу, количестве конкретного товара и т.д. будет храниться в табличной части документа.



Вопрос №10

Проведение документа

Документы служат для сбора и сохранения в базе данных разнообразной информации о хозяйственных событиях в организации. Сбор информации осуществляется путем ввода данных в экранные формы документа, а вот сохранения ее в базе данных осуществляется путем так называемого «проведения документа». Таким образом, проведение документа — это сохранение данных документа в объекты, называемые регистрами. И корректную запись данных в регистры должен обеспечивать такой процесс как **проведение документа**. Документ можно **провести** или просто **записать**. Если документ проведен, то это значит, что он повлиял на учет данных, если же записывается без проведения, то документ с неучтенными данными просто хранится в базе данных.



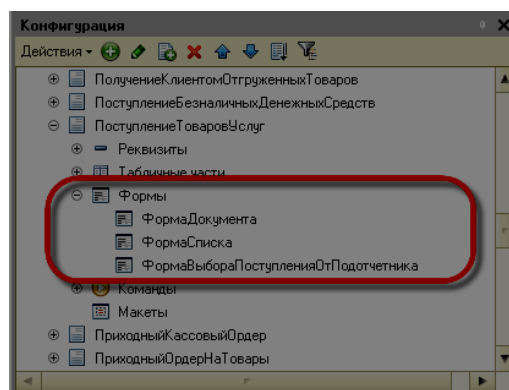
проведения документа.

На рисунке представлен список регистров, участвующих при проведении документа «Поступление товаров и услуг», т.е. этот документ является регистратором для данных регистров. Или еще говорят, «делает движения по данным регистрам».

Конструктор движений — механизм платформы, позволяющий с помощью своего графического интерфейса настроить алгоритм проведения данных документа по указанным регистрам. Результатом работы конструктора является готовая процедура на встроенном языке с именем **ОбработкаПроведения**. Эта процедура располагается в модуле документа и будет вызвана системой в момент

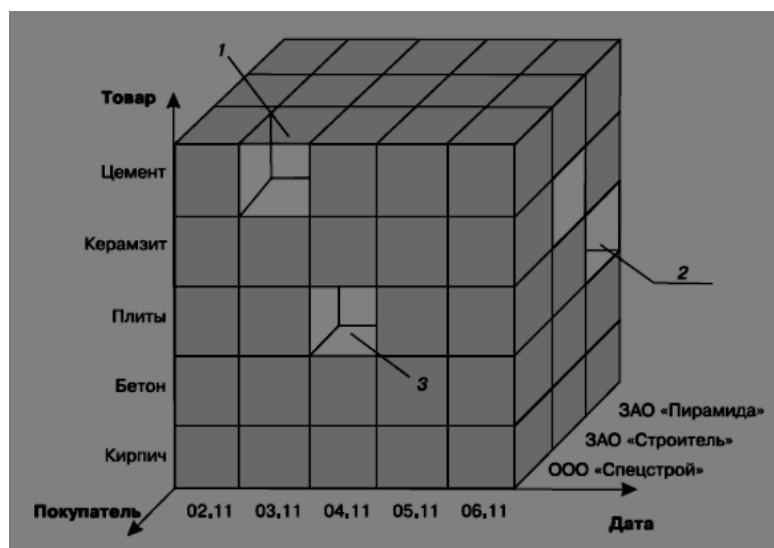
Вопрос №11 Формы документа

Любой документ может иметь несколько экранных форм. Это могут быть ФормаДокумента, предоставляющая интерфейс доступа к отдельному документу, ФормаСписка, предоставляющая доступ к перечню документов информационной базы, ФормаВыбора позволяет сделать выбор отдельного документа в перечне документов. Платформа сама может автоматически генерировать все нужные формы, но разработчик может предоставить пользователю интерфейс, отличный от стандартного.



Вопрос №12 Многомерные хранилища данных

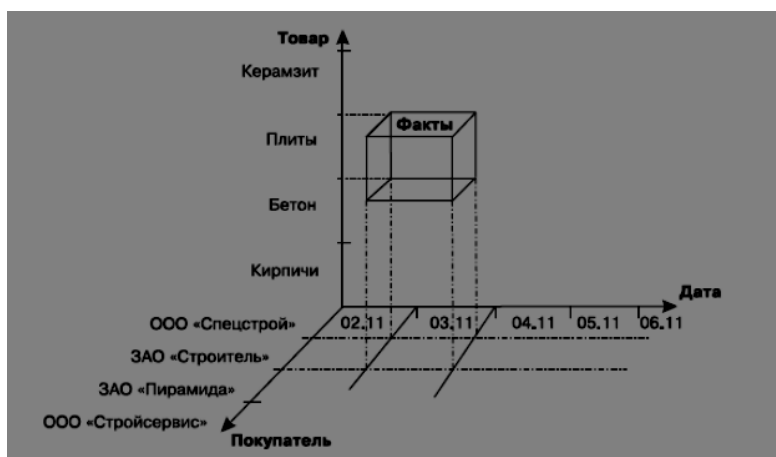
Основное назначение **многомерных хранилищ данных** — поддержка аналитической обработки данных. В основе таких хранилищ лежит многомерная модель данных, опирающаяся на концепцию многомерных кубов или, как еще говорят, гиперкубов. Гиперкубы представляют собой упорядоченные многомерные массивы, которые также часто называют OLAP-кубами (аббревиатура OLAP расшифровывается как On-Line Analytical Processing — оперативная аналитическая обработка). Суть их состоит в следующем. Большинство бизнес-процессов описывается множеством показателей, свойств, атрибутов и т.д. Так, для учета продаж могут понадобиться сведения о наименованиях товаров, поставщике, покупателе, о месте продажи, о ценах, количествах товаров и общих суммах. Для отслеживания процесса во времени может быть использована такая характеристика как дата. Если собрать всю эту информацию в таблицу, то она окажется сложной для визуального анализа и осмысления. Для оперирования такими массивами данных удобно использовать гиперкубы.



Выше представлен пример многомерного куба, который содержит данные в разрезе такой аналитики учета, как Покупатель, Товар, Дата. Т.е. отражает процесс продаж во времени. Каждая продажа в этом случае характеризуется Датой, Покупателем и Товаром. Т.е. один и тот же покупатель может приобретать разные товары у продавца в различные моменты времени. Или один и тот же товар может приобретаться разными покупателями в различные моменты времени и т.д. При этом необходимо учитывать как количество проданного товара, так и цену, по которой он был продан. Таким образом на процесс продаж можно смотреть с различных углов зрения.

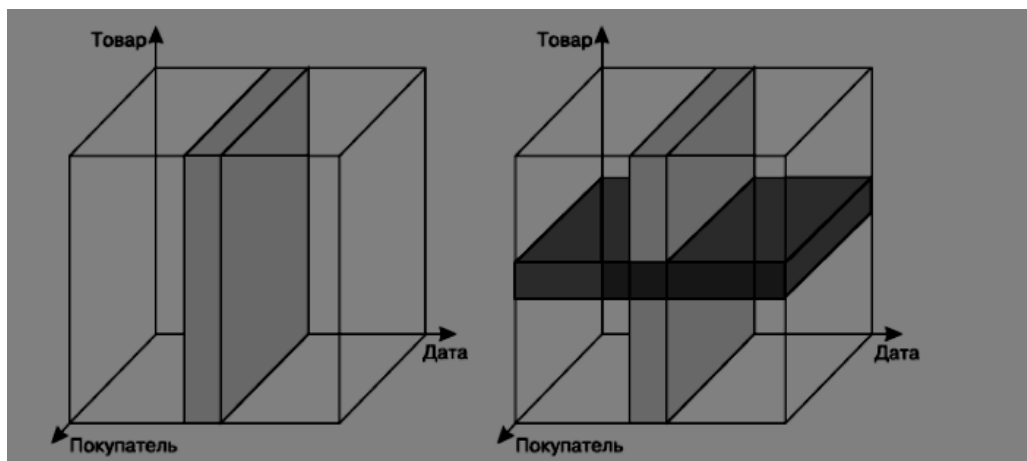
Данный многомерный куб можно рассматривать как систему координат, осями которой являются измерения Дата, Товар, Покупатель. По осям будут откладываться значения даты, наименования товаров, названия покупателей. В

результате получается трехмерный куб данных, каждому набору значений измерений «дата — товар — покупатель» будет соответствовать своя ячейка, в которой можно разместить количество проданного товара данному покупателю в данный момент времени и прочие факты продажи. Таким образом, между процессом продаж его числовыми характеристиками будет установлена однозначная связь. Согласно выше приведенному рисунку гиперкуба в ячейке 1 будут располагаться факты, относящиеся к продаже цемента ООО «Спецстрой» 3 ноября, в ячейке 2 — к продаже плит ЗАО «Пирамида» 6 ноября, а в ячейке 3 — к продаже плит ООО «Спецстрой» 4 ноября.



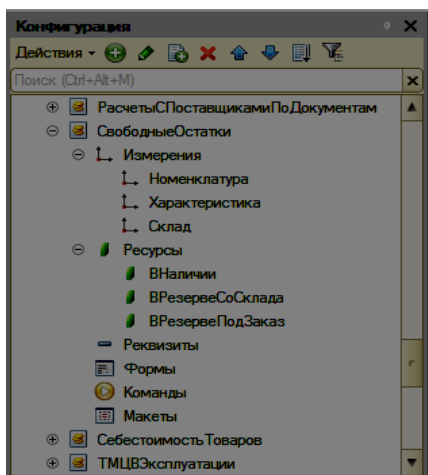
Многомерный взгляд на измерения Дата, Товар и Покупатель представлен на рисунке выше. Количественными фактами, отражающими процесс продаж являются Цена, Количество, Сумма. Выделенная ячейка будет содержать информацию о том, сколько плит, на какую сумму и по какой цене приобрела фирма ЗАО «Строитель» 3 ноября.

В процессе поиска и извлечения из гиперкуба нужной информации над его измерениями производится такая операция как **сечение** или **срез**. Заключается она в фиксации значений одного или нескольких измерений и выделения множества ячеек, соответствующих зафиксированным значениям.



Вопрос №13 Регистры накопления

Учет и хранение данных в 1С:Предприятие базируется на системе регистров. В частности регистров накопления и регистров сведений. По своей сути эти прикладные объекты представляют из себя многомерные кубы данных, измерения которых являются аналитикой учета, а данные, хранящиеся в ячейках такого куба называются ресурсами и количественно характеризуют данную координату ячейку -координату гиперкуба. В данном пункте речь пойдет о двух видах регистров : регистрах накопления и регистрах сведений.



Регистры накопления — это прикладные объекты , обеспечивающие хранение данных во времени, введенных в информационную базу с помощью документов или другими способами в разрезе различной аналитики.

Записи, производящие изменение ресурсов регистра в

базе данных, будут выглядеть следующим образом

Регистр накопления служит для накопления числовых значений. Каждая запись соответствует движению по регистру и выполняет изменение хранимых ресурсов . Движения могут либо

добавлять некоторые приращения к хранимым ресурсам, либо отнимать их. Если должно выполняться увеличение хранимых ресурсов, - такое движение называется движением прихода ("+"), если уменьшение хранимых ресурсов - движением расхода ("-"). Вместе с каждой записью, находящейся в регистре накопления, можно хранить дополнительную произвольную информацию. Для этого служат реквизиты регистра накопления.

Изменение состояния регистра накопления происходит, как правило, при проведении документа, т.е. каждая запись регистра связана с определенным документом - регистратором, номером строки этого документа, и датой — периодом. Состав документов, которые могут создавать записи в регистре накопления, задается разработчиком в процессе создания прикладного решения Система обеспечивает контроль уникальности записей, хранящихся в регистре накопления. Благодаря этому в регистре накоплений не может находиться двух записей, относящихся к одной и той же строке одного и того же документа. Т.е. сочетание значений измерений , периода и регистратора всегда уникально.

Вопрос №14 Виды регистров накопления

Существует два вида регистра накопления — **регистры оборотов и регистры остатков**. **Регистр остатков** позволяет получать остатки ресурсов на определенный период по заданным значениям измерений. **Регистр оборотов** отражает накопление объемов ресурсов за определенный период времени. Типичный пример использования оборотного регистра — регистрация объема продаж. В данном случае нам нужно знать только, какие продажи были за определенный период времени, остатки в данном случае не имеют смысла. Если же цель использования регистра накопления — получение остатков на определенный период, его создают с видом **остатки**. Такой тип позволяет получать как остатки, так и обороты. Для такого регистра система автоматически рассчитывает остатки. Пример «остаточного» регистра — товары на складах, деньги в кассе.

Вопрос №15 Правильное проектирование регистров накопления

Самое трудное — правильно хранить информацию, чтобы её в любой момент было легко получить. Регистры накопления необходимо проектировать из соображений необходимых отчетов.

Использование вида регистра **остатки**, где можно обойтись **оборотами**, считается грубой ошибкой при проектировании регистра накопления с точки зрения производительности системы.

Любая запись регистра делает либо приход «+», т. е. увеличение ресурса по измерениям, либо расход «-», соответственно уменьшение ресурса. Измерения необходимо проектировать так, чтобы по ним происходило как увеличение, так и уменьшение ресурса, в противном случае ресурс либо будет накапливаться до бесконечности, либо убывать, что соответственно приведет к большой нагрузке во время выполнения запросов по данному измерению. И такая ситуация считается грубой ошибкой в проектировании регистров.

Среди особенностей проектирования регистров накопления следует также отметить необходимость правильно расставлять по порядку измерения в регистре.

Выше всего необходимо ставить измерения, которые будут чаще запрашиваться в системе.

Вопрос №16 Индексация измерений регистра накопления

Иногда измерения регистра стоит «индексировать». Т.е. по значениям таких измерений платформа будет создавать индексы для ускорения поиска — упорядоченные по определенному принципу ссылки на записи регистра. Индексируют измерения в тех случаях, когда по ним планируется часто производить отборы при получении данных и данные измерения могут иметь большое количество вариантов значения. Например в регистре «ТоварыНаСкладах» есть два измерения — «Склад, Номенклатура» и один ресурс — «Количество». Правильнее проиндексировать измерение «Номенклатура», а поле «Склад» индексировать не стоит, потому как количество складов в системе, как правило, не существенное.

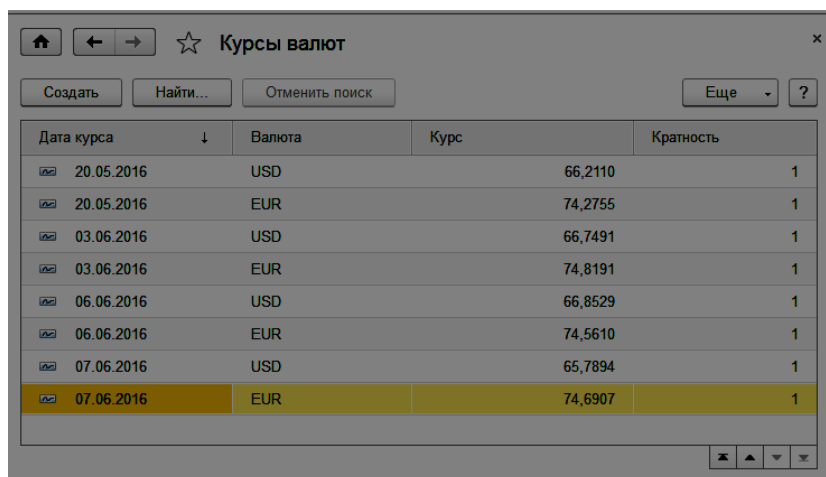
Вопрос №17 Функциональные возможности регистра накопления

Основными функциональными возможностями, которые предоставляет регистр накопления разработчику, являются:

- ☐ выбор записей в заданном интервале по заданным критериям; выбор записей по регистратору;
- ☐ получение остатков и оборотов на указанный момент времени по заданным значениям измерений;
- ☐ отключение использования текущих итогов;
- ☐ расчет итогов на указанную дату;
- ☐ чтение, изменение и запись набора записей в регистр;
- ☐ возможность записи в регистр без пересчета итогов;
- ☐ полный пересчет итогов и пересчет итогов за указанный период.

Вопрос №18 Регистры сведений

Регистр сведений — объект метаданных, предназначенный для хранения справочной информации в разрезе определенных разработчиком измерений, развернутой во времени. Характерный пример использования регистра сведений — хранение информации о курсе валют в разрезе валюты и периода.



Дата курса	Валюта	Курс	Кратность
20.05.2016	USD	66,2110	1
20.05.2016	EUR	74,2755	1
03.06.2016	USD	66,7491	1
03.06.2016	EUR	74,8191	1
06.06.2016	USD	66,8529	1
06.06.2016	EUR	74,5610	1
07.06.2016	USD	65,7894	1
07.06.2016	EUR	74,6907	1

Как и регистр накопления, регистр сведений является многомерным кубом данных, но в отличие от первого предназначен для хранения справочных сведений в разрезе измерений и очень часто отражает изменения таких сведений, развернутые во времени. Очень важным свойством регистра сведений является **периодичность**. Данное свойство позволяет добавить к списку измерений дополнительное измерение — **Период**. Периодичность может принимать следующие значения:

- ❑ Непериодический
- ❑ В пределах секунды
- ❑ В пределах дня
- ❑ В пределах месяца
- ❑ В пределах квартала
- ❑ В пределах года

При выборе периодичности, отличной от варианта Непериодический, система будет контролировать уникальность записей в пределах заданного промежутка времени по совокупности значений измерений. Если запись не уникальна, система выдаст сообщение **Запись с такими ключевыми полями существует!** и не даст произвести запись в базу данных. Одна из главных особенностей периодического регистра сведений — возможность получать готовые значения «Среза первых» и «Среза последних». Эта информация позволяет очень быстро получить из базы данных информацию о последнем (первом) установленном значении на определенную дату.

Регистр сведений может быть независимым либо подчиненным регистратору.

В первом случае записи можно будет произвести как программно, так и из формы списка регистра сведений. Во втором случае обязательно указание документа-регистратора записи. Это накладывает определенные ограничения, но в то же время открывает новые возможности.

Для просмотра данных, содержащихся в регистре сведений, используется форма списка. Она позволяет выполнять навигацию по регистру, добавлять, помечать на удаление и удалять записи регистра. Форма списка позволяет выполнять сортировку и отбор отображаемой информации по нескольким критериям.

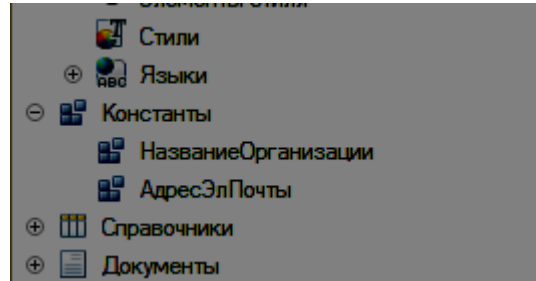
Для просмотра и изменения отдельных записей регистра сведений используется форма записи.

Основными функциональными возможностями, которые предоставляет регистр сведений разработчику, являются:

- создание, изменение и удаление записей;
- выбор записей в заданном интервале по заданным критериям;
- выбор записей по регистратору;
- получение значений ресурсов записей, соответствующих указанному периоду и значениям измерений;
- получение значений ресурсов наиболее ранних и наиболее поздних записей регистра, соответствующих указанному периоду и значениям измерений.

Вопрос №19 Константы

Константы - это прикладные объекты конфигурации, позволяющие хранить в информационной базе данные, которые редко изменяются в течение времени, или не изменяются совсем.



В отличие от справочников константа позволяет хранить только одно значение. Для константы платформа 1С позволяет создать форму ввода значений.

Рис.2 Форма константы

Для константы можно установить имя, синоним, тип данных и т. д.

Рис 3. Палитра свойств константы

Получить значение константы можно с помощью метода Получить(), который вызывается следующим способом:

`Константы.Имя_Нужной_Константы.Получить()` ,

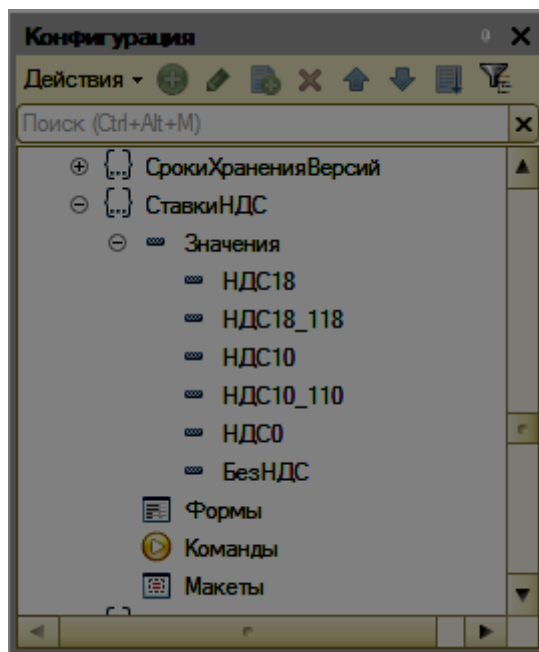
где **Константы.** - обращение к менеджеру констант. Менеджер констант - это объект, позволяющий обращаться и работать с коллекцией констант прикладного решения. С помощью данного объекта можно как обратиться к нужной константе, так и обойти коллекцию в цикле.

Изменить значение константы можно методом Установить().

`Константы.Имя_Нужной_Константы.Установить(Новое_Значение)`

Вопрос № 20 Перечисления

Перечисления — это объекты метаданных, позволяющие хранить в базе данных набор логически связанных однотипных объектов. Например, дни недели, ставки налогов и т. д.



Используются объекты данного типа в основном для подстановки одного из значений набора. Обратиться к конкретному значению перечисления можно следующим способом:

Перечисления.СтавкиНДС.НДС18

Вопрос № 21 **Механизм характеристик**

Механизм описания характеристик - это один из прикладных механизмов платформы, позволяющий описывать произвольным образом элементы справочников, документов и т. д. Так для номенклатуры могут быть введены такие свойства как : цвет, размер, габариты, мощность и т.д. В дальнейшем на основе этих характеристик можно строить отчеты, анализировать объемы продаж, получать другую информацию для принятия решений.

Описание характеристик состоит из таких этапов как: создание характеристик и хранение значений созданных характеристик. Так, чтобы хранить информацию об номенклатуре обуви в разрезе размера, цвета, вида материала, необходимо в информационной базе завести для элементов справочника Номенклатура такие характеристики как : размер, которая будет иметь тип значения Число или СправочникСсылка.Размеры; цвет, которая будет иметь тип значения Строка или СправочникСсылка.Цвета; материал, которая будет иметь тип значения Строка или СправочникСсылка.Материалы. После этого можно указывать их конкретные значения для выбранной номенклатуры: размер - 44; цвет - коричневый; материал - кожа.

Для создания характеристик и их хранения используются такие объекты как :

- План видов характеристик
- Регистр сведений для хранения конкретных значений характеристик, привязанных к конкретной номенклатуре

План видов характеристик

Принципы создания и работы с характеристиками лучше всего рассмотреть на следующем примере: организация занимается производством мебели и учитывает произведенные модели в разрезе материала изготовления, цвета покрытия и иногда формы корпуса, необходимо в прикладном решении отразить возможности такого описания изделий.

Для решения поставленной задачи в конфигурацию можно ввести следующие справочники: Номенклатура для описания изделий, Материалы для описания материала изготовления, Цвета , Формы, а так же подчиненный справочнику Номенклатура справочник **ВариантыНоменклатуры**, хранящий информацию о сочетании значений выше указанной аналитики учета номенклатуры (материал, цвет, форма). Например, Стол может иметь **вариант номенклатуры**: Дуб, Коричневый, Овал, либо :Ольха, Серый, Прямоугольный.

Для хранения аналитики номенклатуры необходимо ввести ПланВидовХарактеристик, в который пользователь в ходе работы может вводить желаемую аналитику.

Справочник `ДополнительныеЗначения` можно создать в конфигурации на случай, если пользователю необходимо будет ввести вид характеристики с типом данных, еще не указанном в списке возможных типов данных. В этом случае значения будут сохраняться в данный справочник и браться из него для дальнейшей подстановки.

Для хранения конкретных вариантов номенклатуры можно ввести регистр сведений `ЗначенияВариантовНоменклатуры` со следующей структурой:

Т.к. справочник `ВариантыНоменклатуры` будет являться владельцем значений вариантов номенклатуры, то для него необходимо указать откуда он будет брать виды характеристик и их значения.

Теперь можно формировать варианты номенклатуры и задавать для них значения.

Вопрос № 22 Механизмы запросов 1С

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

Особенности хранения данных прикладных решений в информационной базе

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



Вопрос № 23 Ссылочные типы данных

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

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

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

Вопрос № 24 Хранение данных ссылочных типов в базе данных

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

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

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

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

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

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

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

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

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

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

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

Вопрос № 25 Хранение данных нессылочных типов

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

The screenshot displays the 1С:Предприятие (1C:Enterprise) interface. On the left is the 'Конфигуратор' (Configurator) tree, where the 'Цены' (Prices) register is selected under 'Регистры сведений' (Information Registers). The main window shows the 'Цены' register configuration with a table of data. Below this, a 'База данных' (Database) window shows the physical table structure for the 'Цены' register.

Цены

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

База данных

Периодический независимый регистр сведений "Цены"

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

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

Вопрос № 26 Источники данных для запросов

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

Таким образом, можно сделать вывод, что все таблицы языка запросов существуют временно в адресном пространстве технологической платформы в отличии от **физических** таблиц базы данных, которые существуют на жестком носителе постоянно. По «степени подобия» физическим таблицам они подразделяются на **реальные** и **виртуальные**. Каждая реальная таблица соответствует конкретной физической таблице. Так, реальная таблица Справочник.Клиенты соответствует справочнику Клиенты, а таблица РегистрСведений.Цены, соответствует регистру сведений Цены. На рисунке ниже представлено сравнение реальной и физической таблиц

Реальные таблицы подразделяются на объектные (ссылочные) и неobjектные (несссылочные). Отличительной особенностью объектных (ссылочных) таблиц является то, что они включают в себя стандартное поле Ссылка, которое позволяет однозначно идентифицировать каждую запись (данные об объекте базы данных). Эти таблицы могут быть иерархическими, подчиненными, и поля таких таблиц могут содержать вложенные таблицы (табличные части). Виртуальные таблицы не похожи на какую-либо физическую таблицу и состоят из других записей, нежели реальные таблицы, т.к. формируются в момент выполнения запроса на основе данных реальных таблиц. Здесь в качестве примера можно привести виртуальные таблицы регистров сведений СрезПоследних , которые содержат информацию о самых последних записях на конкретный период времени по совокупности значений измерений. Т.е. при обращение с запросом к виртуальной таблице СрезПоследних будут предоставлены не все записи физической таблицы регистра сведений, а лишь последние по времени.

Вопрос № 27 Синтаксис текста запросов

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

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

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

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

Вопрос №28 Раздел ИЗ

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

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

Пример 1.

```
ВЫБРАТЬ
    Товары.Код,
    Товары.Наименование,
    Товары.Производитель,
    Цены.Цена
    ИЗ Справочник.Товары КАК Товары ВНУТРЕННЕЕ СОЕДИНЕНИЕ
    РегистрСведений.Цены.СрезПоследних КАК Цены ПО Товары.Ссылка
    = Цены.Товар
```

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

Пример 2.

```
ВЫБРАТЬ
    Товары.Код,
    Товары.Наименование,
    Товары.Производитель,
    Цены.Цена
    ИЗ Справочник.Товары КАК Товары ЛЕВОЕ ВНЕШНЕЕ
    СОЕДИНЕНИЕ      РегистрСведений.Цены.СрезПоследних КАК Цены ПО
    Товары.Ссылка = Цены.Товар
```

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

ВЫБРАТЬ

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

ИЗ Справочник.Товары КАК Товары ПРАВОЕ ВНЕШНЕЕ
СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних КАК Цены ПО
Товары.Ссылка = Цены.Товар

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

ВЫБРАТЬ

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

ИЗ Справочник.Товары КАК Товары ПОЛНОЕ ВНЕШНЕЕ
СОЕДИНЕНИЕ РегистрСведений.Цены.СрезПоследних КАК Цены ПО
Товары.Ссылка = Цены.Товар

Вопрос № 29 Раздел ВЫБРАТЬ

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

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

ВЫБРАТЬ

ЗаказТовара.Клиент КАК Клиент
ИЗ Документ.ЗаказТовара КАК ЗаказТовара

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

ВЫБРАТЬ РАЗЛИЧНЫЕ

ЗаказТовара.Клиент КАК Клиент
ИЗ Документ.ЗаказТовара КАК ЗаказТовара

Клиент
Соколов Иван Андреевич
Орлов Сергей Иванович
Маслова Ирина Николаевна

Вопрос № 30 Раздел ГДЕ

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

ВЫБРАТЬ

 Накладная.Дата КАК Дата,

 Накладная.Номер,

 Накладная.Поставщик

ИЗ Документ.ПриходнаяНакладная КАК Накладная

ГДЕ Дата >= ДАТАВРЕМЯ(2012, 10, 01) И Дата < ДАТАВРЕМЯ(2012, 11, 01)

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

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

Вопрос № 31 Раздел СГРУППИРОВАТЬ ПО

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

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

ВЫБРАТЬ

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

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

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

Товар	Минимум	Максимум	Среднее
Туфли	6 000	8 000	6 750
Сапоги	10 000	12 000	11 000
Кроссовки	3 000	4 000	3 500
Масло	90	100	93.333333
Молоко	50	60	56.666667

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

Вопрос № 32 Раздел ИМЕЮЩИЕ

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

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

ВЫБРАТЬ

НакладнаяСостав.Ссылка.Покупатель КАК Покупатель,
НакладнаяСостав.Товар КАК Товар,
СУММА(НакладнаяСостав.Количество) КАК Количество,
СУММА(НакладнаяСостав.Сумма) КАК Сумма

ИЗ Документ.РасходнаяНакладная.Состав КАК НакладнаяСостав
СГРУППИРОВАТЬ ПО

НакладнаяСостав.Ссылка.Покупатель,
НакладнаяСостав.Товар

ИМЕЮЩИЕ СУММА(НакладнаяСостав.Сумма) > 50000

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

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

Вопрос № 33 Раздел УПОРЯДОЧИТЬ ПО

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

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

ВЫБРАТЬ

Цены.Период КАК Период,

Цены.Товар КАК Товар,

Цены.Цена КАК Цена

ИЗ РегистрСведений.Цены КАК Цены

УПОРЯДОЧИТЬ ПО Период УБЫВ, Цена

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

Вопрос № 34 Раздел ИТОГИ ПО

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

ВЫБРАТЬ

НакладнаяСостав.Товар КАК Товар,
НакладнаяСостав.Количество КАК

Количество,

НакладнаяСостав.Цена КАК Цена,
НакладнаяСостав.Сумма КАК Сумма

ИЗ Документ.ПриходнаяНакладная.Состав КАК
НакладнаяСостав

ИТОГИ

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

ПО Товар

Товар	Количество	Цена	Сумма
Кроссовки	35	3 000	105 000
Кроссовки	15	3 000	45 000
Кроссовки	20	3 000	60 000
Сапоги	10	10 000	100 000
Сапоги	5	10 000	50 000
Сапоги	5	10 000	50 000
Туфли	15	6 500	105 000
Туфли	5	5 000	25 000
Туфли	10	8 000	80 000
Масло	50	75	3 750
Масло	20	80	1 600
Масло	30	70	2 100
Молоко	80	40	3 200
Молоко	30	40	1 200
Молоко	50	40	2 000
Сметана	70	50	3 500
Сметана	50	50	2 500
Сметана	20	50	1 000

Вопрос № 35 Раздел ОБЪЕДИНИТЬ ВСЕ

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

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

ВЫБРАТЬ

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

ИЗ Документ.ЗаказТовара.Состав КАК Заказ
СГРУППИРОВАТЬ ПО

Заказ.Ссылка.Клиент, Заказ.Товар

ОБЪЕДИНИТЬ ВСЕ

ВЫБРАТЬ

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

ИЗ Документ.РасходнаяНакладная.Состав КАК Накладная
СГРУППИРОВАТЬ ПО Накладная.Ссылка.Покупатель,

Накладная.Товар

ИТОГИ ПО Клиент

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

Вопрос №36 Табличный документ и принципы работы с ним

Табличный документ – объект конфигурации, предназначенный для вывода печатных данных на экран, файл и т.д. Обычно табличный документ формируется на основе макета.

Макет представляет собой шаблон для формирования печатной формы. Построение макета производится в конфигураторе в специальном табличном редакторе. Разработчик может создавать области, изменять высоту строк и ширину колонок, задавать форматирование ячеек и всего табличного документа, включать в макет рисунки, диаграммы и сводные таблицы.

Некоторые ячейки макета содержат только текст, который должен быть выведен. Другие ячейки содержат имена параметров (в угловых скобках). Значения этих параметров должны указываться в программном модуле. В ячейку нельзя вписать выражение. Назначение параметров табличного документа должно производиться явно из программного модуля через коллекцию "Параметры" объекта "ТабличныйДокумент".

Общая схема для построения табличного документа (см.рис выше):

1. Создание макета
2. Создание нового табличного документа
3. Получение макета в переменную
4. Получение областей макета
5. Заполнение параметров области
6. Вывод области в табличный документ
7. Установка свойств отображения табличного документа
8. Отображение табличного документа на экране.

Вопрос № 37 Component Object Model (COM)

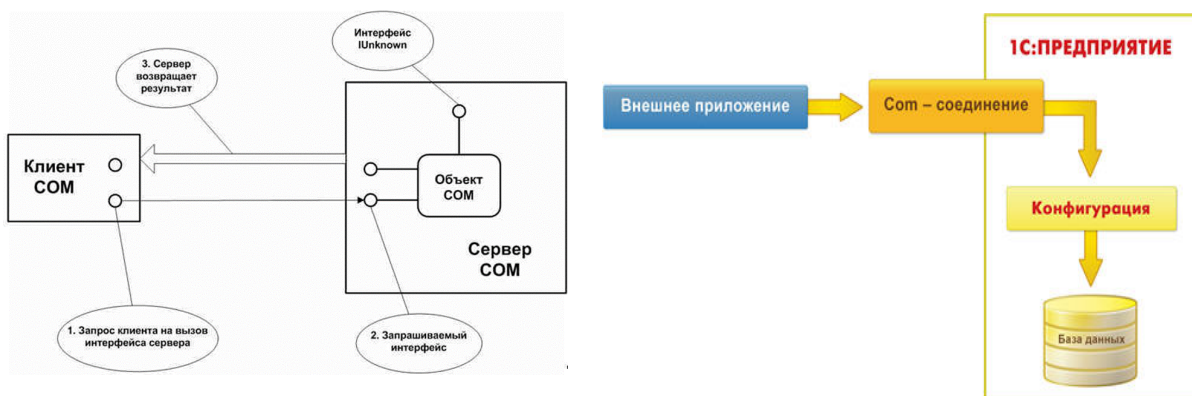
Component Object Model (COM) — это технологический стандарт Microsoft, предназначенный для создания ПО на основе взаимодействующих компонентов объекта, каждый из которых может использоваться во многих программах одновременно. Стандарт COM закрепился в основном на операционных системах семейства Microsoft Windows. В современных версиях Windows COM используется очень широко.

Работает через регистрацию объектов и интерфейсов в реестре

Механизму COM в "1С:Предприятии" отводится ключевая роль в обеспечении взаимодействия прикладных решений с внешними приложениями.

В COM приложение предоставляет для использования свои службы, применяя для этого объекты COM.

Одно приложение содержит как минимум один объект. Каждый объект имеет один или несколько интерфейсов, которые объединяют методы и свойства объекта



Клиент получает доступ к службам объекта только через интерфейс и его методы. Для доступа он должен получить указатель на соответствующий интерфейс. Для каждого интерфейса существует собственный указатель. После этого клиент может использовать службы объекта, просто вызывая его методы. Доступ к свойствам объектов осуществляется только через его методы.

Вопрос №38 Отчеты

Отчет 1С — это специальный объект метаданных, предназначенный для формирования удобного для пользователя представления данных из таблиц базы данных 1С.

Отчеты — это главное в любой информационной системе

Платформа 1С:Предприятие располагает большим количеством конструкторов и механизмов для создания отчетов:

- С помощью макета
- Через механизм «Конструктор выходной формы»
- Через механизм «Система компоновки данных»

Вопрос №39 Система компоновки данных

Система компоновки данных позволяет реализовать следующие возможности:

- - создание отчета без программирования
- - использование автоматически генерируемых форм
- - независимое использование отдельных частей системы компоновки данных;
- - программное управление процессом выполнения отчета.

На рисунке представлена структура СКД:



Настройки компоновки данных описывают все, что может настроить разработчик или пользователь в некоторой установленной схеме компоновки данных. Настройки компоновки данных могут содержать: отбор, упорядочивание, условное оформление, структуру отчета; параметры получения данных; параметры вывода данных.

Схема компоновки данных описывает суть данных, которые предоставляются отчету. Представляет собой базу, на основе которой могут быть сформированы всевозможные отчеты. Схема компоновки данных может содержать: текст запроса с инструкциями системы компоновки данных; описание нескольких наборов данных; подробное описание

доступных полей; описание связей между несколькими наборами данных; описание параметров получения данных; описание макетов полей и группировок;

Макет компоновки данных представляет собой уже готовое описание того, как должен быть сформирован отчет. В нем соединяется схема компоновки и настройки компоновки. Фактически макет компоновки данных представляет

собой результат применения конкретных настроек к схеме компоновки и является готовым заданием процессору компоновки на формирование отчета нужной структуры с учетом конкретных настроек.

Результат компоновки данных представляется набором элементов результата компоновки данных. Как самостоятельная логическая сущность результат компоновки данных не существует, существуют только его элементы. Элементы результата компоновки данных можно вывести в табличный документ для представления их пользователю, или в другие виды документов. Также имеется возможность программного вывода элементов результата компоновки в объекты вида Дерево значений или Таблица значений.

Вопрос № 40 Схема компоновки данных

Основой компоновки отчета является Схема компоновки данных, базовой составляющей которой является Набор данных. Набор данных может быть создан на основе запроса, объекта в оперативной памяти и объединения наборов.

Ресурсы — это групповые итоговые данные. Для вывода ресурсов обязательно необходимы группировки. На основании значений ресурсов формируются общие итоги отчета. В таблице ресурсы выводятся на пересечении соответствующих строк и колонок. В качестве значений диаграммы выводится первый ресурс отчета.

Параметры используются для задания получения конкретных интересующих пользователя сведений. Например, итоги на конкретную дату и т. д.

Структура отчета может задаваться в виде списка, диаграммы, таблицы

Условное оформление позволяет оформить некоторым стилем определенные строки или ячейки отчета.

Вопрос №41 Задача

С помощью платформы 1С: Предприятие 8 создать ПСОД “Карточка читателя библиотеки”. Посредством данной ПСОД библиотекарь должен иметь возможность хранить данные о выданных и сданных книгах читателями. Информация о книгах библиотечного фонда и читателях должна храниться в справочниках. История пользования книгами читателем: (дата, читатель, книга, количество, дата начала пользования, дата окончания пользования, вид операции (сдача или выдача)), должна храниться в регистрах и вводиться документом. Библиотекарь на основе созданного документа должен иметь возможность распечатать карточку выданных и сданных книг за текущее посещение.

Вопрос №42 Задача

С помощью платформы 1С: Предприятие 8 создать ПСОД “Рецептурная карта блюда”. Посредством данной ПСОД составитель меню должен иметь возможность хранить данные об ингредиентах, входящих в блюдо: название продукта, его вес, калорийность продукта, рассчитанная исходя из данных о калорийности продукта на 100 г и рассчитанная пропорционально его весу в блюде. А также общую калорийность блюда. Информация о названиях блюд хранится в справочнике. Информация о продуктах и их калорийности хранится в справочнике. Рецепттурная карта создается документом и ее данные хранятся в регистре. На основе созданного документа составитель должен иметь возможность распечатать рецепттурную карту.

Вопрос №43 Задача

С помощью платформы 1С: Предприятие 8 создать ПСОД “Парикмахерские услуги”. Посредством данной ПСОД администратор парикмахерской должен иметь возможность вводить и хранить данные о клиенте и оказанных ему услугах в разрезе их стоимости, а также общую стоимость обслуживания клиента. Информация об оказанных клиенту услугах вводится документом и хранится в регистре. Данные о клиенте, видах работ и ценах на них хранятся в справочниках и выбирается из них в документ. Из документа должна быть реализована возможность печати перечня оказанных клиенту услуг, ценах на них и общей стоимости обслуживания.