

Конфигурирование в системе
«1С:Предприятие 8». Решение
оперативных задач

Основная цель данного курса

- Получить навыки самостоятельной работы по созданию оперативных учетных и управленческих решений.

Объектная схема построения конфигураций

- Документ – Основа любой системы учета. потому как документ – объект, предназначенный для регистрации событий, важных с точки зрения некой предметной области.

Объектная схема построения конфигураций

- Отчеты – класс объектов для обобщения, формирования и вывода в удобном для пользователя виде регистрируемой входящей информации (документов).

Объектная схема построения конфигураций

Регистры учета – получаем средство быстрого формирования отчетности за счет предварительного накопления и подготовки отчетных данных.

- Регистры сведений;
- Регистры накопления;
- Регистры бухгалтерии;
- Регистры расчета.

Объектная схема построения конфигураций

Создавая Регистры мы намеренно вносим в систему избыточность. Как правило, в регистры попадают данные из документов а не от пользователя. Но за счет регистров неимоверно возрастает быстрота и скорость получения отчетной информации. При чем можно учитывать плановые операции(т.е. те что должны произойти в будущем)

Объектная схема построения конфигураций

Классы справочных объектов служат для организации правильного заполнения документов.

Справочные объекты:

- Справочники;
- Перечисления;
- Константы;
- Планы счетов;
- Планы видов характеристик;
- Планы расчетов;

Роль и место регистров

- Корень автоматизации любой учетной системы: система обязана максимально быстро, максимально точно быть готовой предоставлять информацию по тем показателям, состоянбике которых ей надлежит регистрировать
- Отчетность – средство обеспечения существования бизнеса (регламентированная отчетность) и средство анализа.

Природа показателей

- Для любого явления можно придумать сотни и тысячи показателей.
- По Законам Паркинсона: Любая бюрократическая система естественным образом увеличивает количество контролируемых показателей экспоненциально с течением времени, вплоть до момента «удушения» дела, которым полагалось управлять.

Показатели

- Первая группа – «показатели остатка» - используются для учета явлений, которые то, прирастают то убывают, но нам необходимо знать их состояние на последний (или на каждый) момент времени.

Текущее состояние показателя остатка тесно связано с прошлым их состоянием.

Показатели

- Вторая группа – «показатели оборотные» -характеризуют движение за какой-то период. Учитываются явления растущие только в одну сторону.



Наличие периода в определении



Независимость его состояния за текущий период от состояния за прошлый период.

Показатели

- Третья группа – «показатели состояния» (разовые явления, устанавливающие, показатели в определенное состояние)

Устанавливаемое состояние не зависит в принципе от прошлого состояния, но будет действовать в рамках какого-то жесткого периода.

Задача любой учетной или управленческой системы

Мгновенно выдавать информацию по состоянию того или иного показателя.

Мгновенно – сводя к минимуму затраты времени на расчеты значения.

Регистры – объекты, предназначенные для хранения и практически мгновенной выдачи значений показателей в произвольных разрезах.

Структура регистров

- Ресурсы – хранят данные регистра, то есть та информация, ради которой был создан регистр.
- Измерения – используются для обеспечения разрезов этой информации.
- Реквизиты – используются для хранения информации о записях регистра.

При проектировании структуры регистров

- Измерениями регистров могут быть объекты ссылочных или примитивных типов данных
- Количество измерения определяет размерность регистров и следовательно от него зависит физический размер таблиц регистра

Понятия связанные с регистрами

- Понятие Граница периода рассчитанных итогов связано с внутреннем порядком хранения данных в реальных таблицах регистра



Атрибуты регистров

- Регистратор – атрибут, содержащий ссылку на документ, изменивший состояние ресурсов регистра.
- Период – атрибут, указывающий на период (момент времени), с которого действуют данные изменений ресурсов
- Вид Движения

При включенном использовании итогов и текущих итогов регистра:

- При любом изменении записей в таблице движений – пересчитываются актуальные итоги (на 01.11.3999)
- Пересчитываются все промежуточные итоги, у которых период итога больше периода измененных записей движений
- При задействовании системных функций управления итогами (установка периода рассчитанных итогов, пересчет итогов.

Резюме

- При получении итогов система всегда посчитает их правильно, но чем ближе к реальной дате граница периода рассчитанных данных в таблице итогов – тем быстрее система сама будет выдавать требуемые данные.

Управление Итогами

- Программно (Синтаксис-помощник
Прикладные объекты/ регистры
накопления/
РегистрыНакопленияМенеджер<Имя
регистра накопления>/ Методы)
- В пользовательской части программы

Управление Итогами

- Есть возможность управлять установкой режима разделения итогов. Данная возможность повышает параллельность работы в системе (Сеансам в случае конкуренции при работе с итогами регистров не надо дожидаться освобождения ресурсов), но принципиально может приводить к существенному разрастанию таблицы итогов. Ведь сворачивание выполняется только при пересчете итогов.

Режим разделения итогов

- Для включения данного режима необходимо установить для регистра флаг «Разрешить разделение итогов», а далее в пользовательском режиме «Включить разделение итогов»

Агрегаты

- Это средство оптимизации времени доступа к данным оборотных регистров за счет за благовременного хранения в БД доп. таблиц с различными вариантами обобщения ресурсов в разрезе комбинации измерений и стандартных интервалов (помесечно, понедельно, подекадно, поквартально, по годам и т.д.)

Использование объектной модели

Чтобы получить накопившиеся итоги в регистре остатков, достаточно воспользоваться методом объекта «РегистрНакопленияМенеджер» - Остатки().

Синтаксис:

Остатки(<Момент времени>, <Отбор>, <Измерения>, <Ресурсы>)

Использование объектной модели

Параметры:

- <Момент времени> (необязательный) - Тип: Дата<Момент времени> (необязательный) - Тип: Дата, МоментВремени<Момент времени> (необязательный) - Тип: Дата, МоментВремени, Граница<Момент времени> (необязательный) - Тип: Дата, МоментВремени, Граница. Момент времени, на который необходимо получить остатки. Если параметр не указан, то будут получены текущие остатки (на максимальную дату движений регистра). Если в качестве параметра передана Дата<Момент времени> (необязательный) - Тип: Дата, МоментВремени, Граница. Момент времени, на который необходимо получить остатки. Если параметр не указан, то будут получены текущие остатки (на максимальную дату движений регистра). Если в качестве параметра передана Дата, то остатки будут получены на начало дня, если МоментВремени - то в остатках не будут учтены движения регистра, произошедшие в тот же момент времени.
- <Отбор> (необязательный) - Тип: Структура. Структура, содержащая набор значений для измерений регистра. Имя ключа структуры должно совпадать с именем измерения

Использование объектной модели

- <Измерения> (необязательный) - Тип: Строка. Список измерений, для которых надо получить остатки. Строка, содержащая имена измерений, разделенные запятыми. Если параметр не указан или указана пустая строка, то остатки будут сформированы по всем измерениям.
- <Ресурсы> (необязательный) - Тип: Строка. Список ресурсов, для которых надо получить остатки. Строка, содержащая имена ресурсов, разделенные запятыми. Если параметр не указан или указана пустая строка, то остатки будут сформированы по всем ресурсам.
Значение по умолчанию: Пустая строка

Использование объектной модели

Результат выполнения метода –
Таблица значений, заполненная
итогами и содержащая колонки с
измерениями, указанными в
парамetre «Измерения», и колонки с
ресурсами, указанными в параметре
«Ресурсы»

Основные причины применения объектной модели обращения к БД

- Когда прикладной объект надо записывать в БД
- Когда прикладной объект надо модифицировать (прочитать, изменить и записать)
- Когда в алгоритме нужно использовать механизм динамического чтения данных
- Когда данные уже лежат в оперативной памяти

Использование табличной модели

Табличная модель системы «1С: Предприятие 8» - это получение данных через обращение к таблицам базы данных с помощью **языка запросов**.

Технология составления схемы запроса

- Нарисовать эскиз выходной таблице запроса с вариантами заполнения данных
- Обозначить на эскизе количество выходных полей
- Обозначить для каждого поля таблицу-источник
- Обозначить дополнительные действия в запросе

Дополнительные действия:

- Фильтрация, отбор
- Свертка таблицы (СГРУППИРОВАТЬ ПО)
- Разворачивание таблицы за счет добавления строк с промежуточными итогами (ИТОГИ ПО)
- Сортировка (УПОРЯДОЧИТЬ ПО)

Использование табличной модели

- Результатом как при обращении через табличную модель будет таблица вида:

РегистрНакопления.Имя.Остатки(Период,
Условие);

| Поле | Тип |
|-----------------|---------------|
| <Измерение> | Тип измерения |
| <Ресурс>Остаток | Число |

Таблица получения Оборотов

Синтаксиз:

РегистрНакопления.<ИмяРегистра>.
Обороты(<Начало периода>, <Конец
периода>, <Периодичность>,
<Условие>)

Таблица получения Оборотов

Параметры:

- <Начало периода> (необязательный) - Тип: Дата<Начало периода> (необязательный) - Тип: Дата, МоментВремени<Начало периода> (необязательный) - Тип: Дата, МоментВремени, Граница<Начало периода> (необязательный) - Тип: Дата, МоментВремени, Граница. Момент времени, начиная с которого необходимо получить обороты. Если параметр не указан, то обороты будут получены начиная с момента ведения базы данных. Если в качестве параметра передана Дата<Начало периода> (необязательный) - Тип: Дата

Таблица получения Оборотов

- <Конец периода> (необязательный) -Тип: Дата<Конец периода> (необязательный) - Тип: Дата, МоментВремени<Конец периода> (необязательный) -Тип: Дата, МоментВремени, Граница<Конец периода> (необязательный) -Тип: Дата, МоментВремени, Граница. Момент времени, определяющий конец периода за который необходимо получить обороты. Если параметр не указан, то обороты будут получены до самого позднего движения регистра. Если в качестве параметра передана Дата<Конец периода> (необязательный) -Тип: Дата

Таблица получения Оборотов

- <Периодичность> (не обязательный) - используется для указания дополнительного разворота итогов по периодичности.

Может принимать значения:

- Период – Тогда итоги не разворачиваются
- Регистратор – по документу регистратору.
- день – по дням
- Неделя, - Месяц, Квартал, Год – соответственно.

Таблица получения Оборотов

- В качестве <Условия> может выступать логическая конструкция, оперирующая с полями регистра накопления. Используется для ограничения состава записей, по которым будут выбираться итоги.
- При получении итогов выходной таблицы будут использоваться только активные записи регистра.

Таблица получения Оборотов

РегистрНакопления.Имя.Обороты(Период, Условие);

| Поле | Тип |
|----------------|--------------------|
| Период | Дата |
| Регистратор | ДокументСсылка.Имя |
| <Измерение> | Тип измерения |
| <Ресурс>Оборот | Число |
| <Ресурс>Приход | Число |
| <Ресурс>Расход | Число |

Таблица получения Остатков и Оборотов

Синтаксиз:

РегистрНакопления.<ИмяРегистра>.

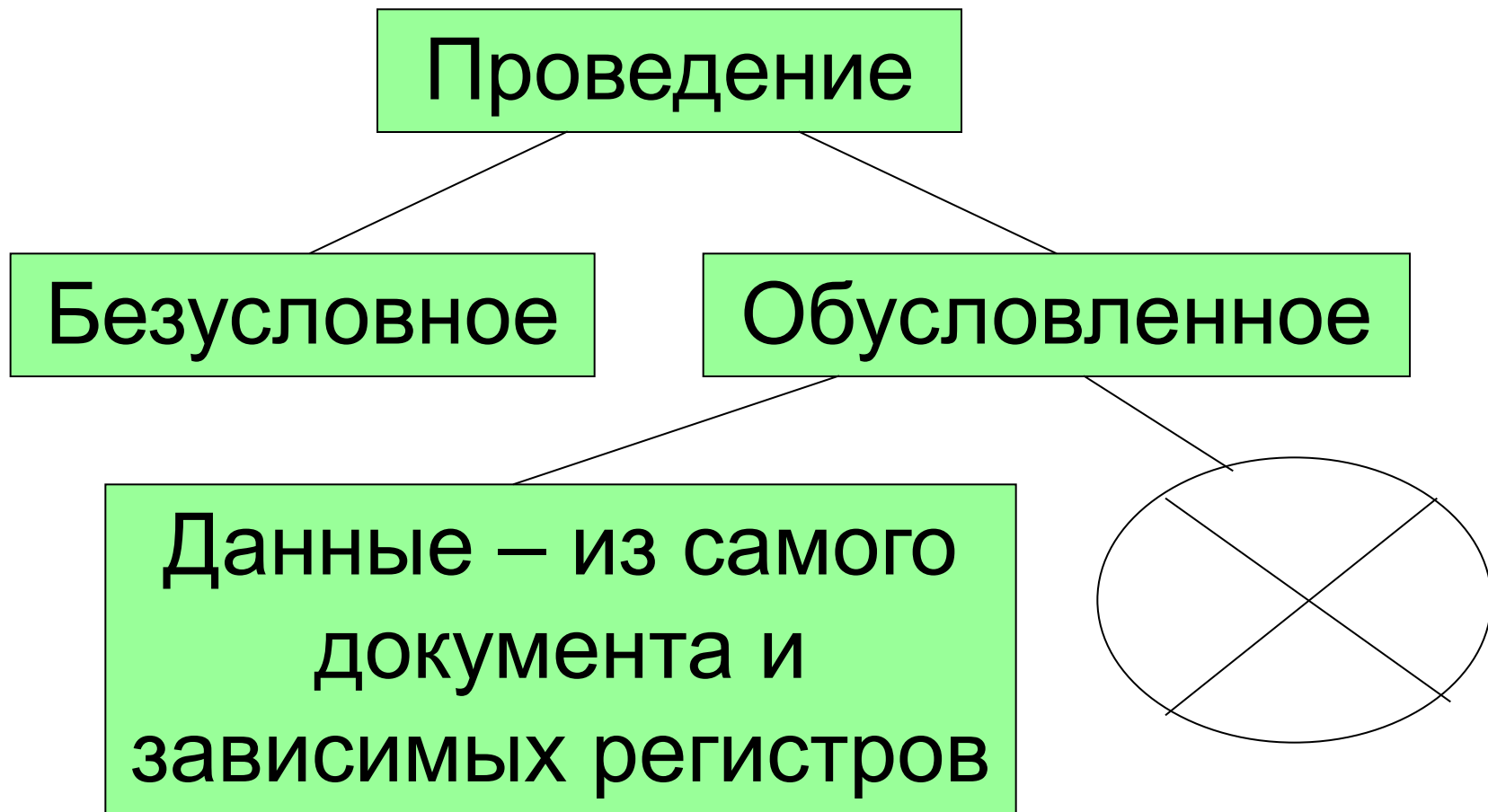
ОстаткиИОбороты(<Начало периода>, <Конец периода>, <Периодичность>, <МетодДополнения>, <Условие>)

<МетодДополнения> - параметр задающий, надо или нет показывать периоды, соответствующие значениям <НачалоПериода> и <КонецПериода>, если в них не было никаких движений.

Формирование отчетов по средствам СКД

- Начиная с версии 8.1 платформы – система «1С:Предприятия» представляет к использованию такое мощное средство **Декларативного** формирования отчетности (достаточно лишь визуальных средств для объяснения системе – что хотелось получить), как «Схема компоновки данных».

Обусловленное провидение



Сборка алгоритма проведения документа

1. Конструктор движений
2. Обозначить недостающие действия комментариями
3. Определяем обусловленность проведения
4. Составляем текст запроса по недостающим данным (При обусловленном проведении)

Сборка алгоритма проведения документа

5. Обработка текста запроса в консоли запросов
6. Сборка текста запроса должна выполняться по шагам, от простого к сложному с проверкой после каждого шага.
7. Обеспечить максимальное быстродействие запроса для конкретной задачи

Сборка алгоритма проведения документа

8. Готовый текст запроса использовать для сборки фрагмента процедуры посредством конструктора запроса с обработкой результата.
9. Отладка
10. Обеспечение быстродействия всей процедуры в целом.

При написании запроса следует ПОМНИТЬ:

- Соединение – это попытка в одном запросе получить всевозможные комбинации записей исходных таблиц, ограниченная лишь количеством и видами связей
- Связь – условие, которому обязательно должна соответствовать очередная комбинация, которую пытаются построить соединение

При написании запроса следует ПОМНИТЬ:

- Если связей не накладывать вообще, то будем получать «декартово произведение» комбинаций записей (каждая - к каждой). Поэтому:
«Минимальное количество связей на единицу меньше количества таблиц источников»

При написании запроса следует ПОМНИТЬ:

- Чаще всего можно определить ведущий источник и ведомый источник. Выходные поля группировочных сущностей должны определяться из ведущих источников

Оперативное проведение

- Для любого реального объекта существует 3 понятия времени: «Прошлое», «Настоящее» и «Будущее».
- Идеальная система учета, как отображения реальности, стремится фиксировать события в момент их происхождения – т.е. в реальном времени.

Оперативное проведение

- При Оперативном проведении док-тов система позволяет добиться расстановки документов в идеальной хронологической последовательности.
- С точки зрения отражения док-та в учете сам механизм ОП не выполняет никаких действий, он только передает параметра в обработчик проведения.

Блокировка записей регистров

По умолчанию выполнение запроса по любому объекту происходит в режиме чтения данных.

- В файл-серверном варианте использования программы при проведении док-та накладывается блокировка на таблицы целиком.

Блокировка записей регистров

- При клиент-серверном варианте использования – блокировки накладываются на диапазоны записей таблиц, а не на таблицы целиком.
- **Запрос.Текст = Запрос.Текст + «**
| **ДЛЯ ИЗМЕНЕНИЯ**
| **РегистрНакопления.**
| **ОстаткиНоменклатуры.Остатки»;**

Возможные коллизии при проведении.

- Проводя что-то задним числом пользователь может легко нарушить логику формирования и проведения последующих документов.

Способы решения коллизий

- Последовательность документов – средство группирования документов в едином хронологическом порядке для обеспечения ведения единой логики событий в рамках некоего механизма учета.

Правила внесения изменений в структуру регистров «Живой» базы

- 1. Продумать все детали механизма (какие структурные измерения будут в регистрах (+/-)).
- 2. Внести структурные измерения в регистр. (чем больше измерений, тем дольше выполняются запросы).
- 3. Прописать положительные движения по данному регистру.

Правила внесения изменений в структуру регистров «Живой» базы

- 4. Отладить написанный алгоритм проведения для одного документа, но для каждого вида документа, который делает «+»- движения.
- 5. Прописать «Отрицательные» движения (Если требуется)
- 6. Отладить прописанное на проведении для одного документа.

Правила внесения изменений в структуру регистров «Живой» базы

- 7. Убедиться в корректности работы всех механизмов, кот. базируются на этом регистре.
- 8. Перепроверите все документы, связанные с данным регистром.

В случае невозможности проведения старых документов по новым законам придется реализовать корректное использование нужного механизма.

Реализация алгоритмов проведения в ситуациях с повышенными требованиями к быстродействию системы

- Абсолютная эффективность системы – мера, степень соответствия полученных результатов задекларированным целям.
- Относительная эффективность – мера, степень соответствия полученных результатов к затратам, расходам, обусловившим, обеспечившим получение этих результатов.

Реализация алгоритмов проведения в ситуациях с повышенными требованиями к быстродействию системы

- Разработка эффективной системы – это всегда нахождение баланса между достижением достаточно эффективных результатов по каждой из целей системы.
- Идеальных с точки зрения эффективности алгоритмов быть не может в принципе.

Реализация алгоритмов проведения в ситуациях с повышенными требованиями к быстродействию системы

- Нахождение достаточно эффективного алгоритма – это творческий процесс конструирования сбалансированной системы достижения этих целей.

Направления оптимизации алгоритмов

- Минимизация количества обращений к базе данных.
- Минимизация объема данных, обрабатываемых в процессе проведения (а именно: минимизация количества таблиц ИБ, задействованных в алгоритме; минимизация размеров обрабатываемых таблиц; минимизация действий с таблицами БД)

Направления оптимизации алгоритмов

- Минимизация количества и времени действия блокировок;
- Оптимизация сценариев обработки данных в плане приоритета решений, ориентированных на повышение параллельности работы перед ориентированными на дополнительный сервис пользователей

Приемы для повышения эффективности системы:

- I. Применить разделение обработки отражения документов в учете на 2 части:
 - I. Действия, требующие секундной обработки в момент проведения документа;
 - II. Действия, которые можно отложить на выполнение в более спокойный период (в конце смены, ночью, только перед формированием регламентированной отчетности)

Приемы для повышения эффективности системы:

II. Можно применять структурное решение проблемы времени расчета показателей. (сохранять часть рассчитанных в процессе показателей в промежуточных регистрах. Или ввести служебные показатели, которые в отчетах в последствии не используются, а только помогают быстрее проводить документы).

Приемы для повышения эффективности системы:

III. Можно сократить время действия блокировок, накладываемых в процедуре проведения за счет сдвига момента наложения этих блокировок как можно ближе к концу алгоритма.

Приемы

- Исключим вопрос списания себестоимости из секундных;
- Введем регистр «Свободные Остатки» (Количество В Свободном Остатке, Номенклатура, Склад) для контроля отриц. остатков;
- Поменяем «Чтение из регистра\Запись в этот регистр» на «Запись в этот регистр\ Чтение из регистра»

Это даст следующие плюсы:

- Не придется возиться с регистром «ОстаткиНоменклатуры» ни на запись ни на чтение;
- Ни в один набор записей не придется записывать набор записей дважды;
- Блокировки будут действовать меньше времени

Это даст следующие плюсы:

- Запрос для контроля остатков будет строиться только по одному регистру с актуально рассчитанными остатками
- Запрос для контроля отрицательных остатков еще больше можно ускорить, заложив ситуацию перепроведения документа.(контроль разницы)
- Расчет себестоимости выносится в отдельный механизм, который будет запускаться регламентно.

Универсальный алгоритм проведения документов

- Запрос по шапке
- Определение структуры проведения
- Проверка заполнения реквизитов шапки и условий возможности проведения
- Формирование структуры параметров проведения
- Формирование движений по шапке документа (пока в наборе записей, но не в регистре)

Универсальный алгоритм проведения документов

- Запрос по таб. Части с получением данных из других объектов
- Проверка заполнения реквизитов в строке табличной части и условий возможности проведения
- Формирование движений по строке таб. Части
- Запись наборов движений по регистрам

Правила работы с регистрами остатков:

- **Итоги каждого ресурса рано или поздно должны иметь возможность выводиться «В НОЛЬ»!**
- **Пользуясь ресурсами – нельзя учитывать показатели движения!**
- **Нельзя по какому-либо из ресурсов делать рассогласованные по набору измерений движения!**
- **Для учета характеристик движения можно пользоваться «Реквизитами» регистра!**

Получение последних значений регистра сведений

- ПолучитьПоследнее() – Возвращает данные в виде объекта Структура, содержащего значения ресурсов.
- СрезПоследних()-Возвращает данные в виде объекта ТаблицаЗначений, содержащего как значения ресурсов, так и значения измерений.

Получение последних значений регистра сведений

РегистрСведений.Имя.СрезПоследних(Дата, Условие);

| Поле | Тип |
|-------------|--------------------|
| Период | Дата |
| Регистратор | ДокументСсылка.Имя |
| НомерСтроки | Число |
| <Измерение> | Тип измерения |
| <Ресурс> | Тип ресурса |
| <Реквизит> | Тип реквизита |