

Практична робота № 8

Тема: оптимізація проведення документу «Надання послуг» у програми 1С:Підприємство.

Мета: ознайомитися з реєстрами накопичення у програмі 1С:Підприємство 8.2 . Навчитися створювати простий звіт у програмі 1С:Підприємство 8.2.

МЕТОДИЧНІ ВКАЗІВКИ

У цьому розділі ми поговоримо про особливості використання довідкових даних, так як, використовуючи доступ до цих даних за допомогою запитів, ми можемо значно підвищити швидкість проведення документа і оптимізувати цей процес.

Терміном посилальні дані ми будемо позначати дані, що зберігаються в базі даних, доступ до яких можливий за допомогою об'єктів вбудованої мови виду Посилання: ДвідникСилка. <Ім'я>, ДокументСилка. <Ім'я> і т. д. Для того щоб подальший виклад було зрозуміліше, ми побудуємо пояснення на прикладі отримання посилання на вид номенклатури при проведенні документа НаданняПослуг.

Не всі дані, що зберігаються в базі даних, є посилальними. Це пов'язано з тим, що в моделі даних «1С: Підприємство 8» існує поділ на дані, що представляють об'єктні суті (довідники, плани рахунків, документи і т. Д.), І дані, що представляють Необ'єктні суті (реєстри відомостей, реєстри накопичення і т. д.).

З точки зору платформи деяка сукупність об'єктних даних визначається не тільки значеннями своїх полів, а й самим фактом свого існування. Іншими словами, видаливши з бази деяку сукупність об'єктних даних, ми не зможемо повернути систему в той же стан, який був до видалення. Навіть якщо ми заново створимо ту ж саму сукупність об'єктних даних з тими ж самими значеннями полів, з точки зору системи це буде ІНША сукупність об'єктних даних.

Кожну таку сукупність об'єктних даних, унікальну з точки зору системи, називають об'єктом бази даних.

Для того щоб система могла відрізнити один об'єкт бази даних від іншого, кожен об'єкт бази даних (сукупність об'єктних даних) має внутрішній ідентифікатор. Різні об'єкти бази даних завжди матимуть різні внутрішні ідентифікатори. Цей ідентифікатор зберігається разом з іншими даними об'єкта в спеціальному полі Посилання.

Необ'єктні дані зберігаються у вигляді записів і з точки зору системи визначаються виключно значеннями своїх полів. Таким чином, видаливши деякий запис і записавши після цього нову, з точно такими ж значеннями всіх полів, ми отримаємо той же самий стан бази даних, яке було до видалення.

Таким чином, оскільки ми можемо однозначно вказати на кожен об'єкт бази даних, у нас з'являється можливість зберігати такий покажчик в полях інших таблиць бази даних, вибирати його в поле введення, вказувати в параметрах запиту під час пошуку по посиланню і т. д.

У всіх цих випадках якраз і буде використовуватися об'єкт вбудованої мови виду Посилання. Фактично цей об'єкт зберігає тільки внутрішній ідентифікатор, що знаходиться в полі Посилання.

Наприклад, якщо взяти наш документ НаданняПослуг, то в поле, що зберігає реквізит табличної частини Номенклатура, насправді знаходиться внутрішній ідентифікатор, який вказує на елемент довідника Номенклатура



Коли в обробнику події ОбробкаПроведення документа Надання Послуги ми присвоюємо значення реквізиту табличній частині Номенклатура будь-якої змінної, ми маємо справу з об'єктом вбудованої мови ДокументОб'єкт.НаданняПослуг.

Цей об'єкт містить в собі значення всіх реквізитів документа і реквізитів його табличних частин.

Тому звернення (лістинг 8.1) призводить до того, що ми просто читаємо дані, що зберігаються в оперативній пам'яті, в цьому самому об'єкті вбудованої мови (рис.8.2)

```
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
```



Однак коли ми звертаємося до виду номенклатури як до реквізиту того елемента довідника, посилання на який вказана в табличній частині документа (лістинг 8.2), відбувається буквально наступне (рис. 8.3).

```
Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =  
Перечисления.ВидыНоменклатуры.Материал Тогда
```



Оскільки в об'єкті ДокументОб'єкт.НаданняПослуг є тільки посилання на елемент довідника Номенклатура і більше ніяких даних про цей елемент немає, платформа візьме це посилання і звернеться по ній в кеш об'єктів в надії знайти там дані того об'єкта, посилання на який у неї є.

Якщо кеш об'єктів не матиме потрібних даних, він звернеться до бази даних з тим, щоб прочитати всі дані об'єкта, посиланням на який він має.

Після того як всі дані, що зберігаються в реквізитах потрібного елемента довідника і в реквізитах його табличних частин, будуть прочитані в кеш об'єктів, кеш об'єктів поверне запитувану посилання, що зберігається в реквізиті ВидНоменклатури довідника Номенклатура.

Як нескладно здогадатися, подібне звернення до бази даних вимагає більшої кількості часу, ніж просто читання з оперативної пам'яті. При інтерактивному заповненні документа подібні затримки мізерно малі, в порівнянні зі швидкістю роботи користувача. Однак при виконанні великої кількості розрахунків (наприклад, при проведенні великих документів, що містять кілька тисяч рядків) різниця в часі може бути досить помітною.

З усього вищесказаного можна зробити наступний висновок: якщо алгоритм проведення документа використовує тільки ті дані, які присутні в реквізитах документа (і його табличних частин), цілком достатньо використовувати конструктор рухів документа (як це було у нас у випадку з документом Прибуткова Накладна).

Якщо ж в алгоритмі проведення потрібно аналізувати додаткові реквізити об'єктів, посилання на які містяться в документі, а також використовувати результати розрахунку підсумків реєстрів, слід використовувати запити для більш швидкої вибірки даних з бази даних.

Те ж саме справедливо по відношенню до виконання будь-яких ділянок програми, критичних по продуктивності. Механізм запитів краще «читає» інформаційну базу і може за один раз вибрати тільки ті дані, які необхідні. Тому, наприклад, в типових рішеннях ви практично не побачите використання об'єкта вбудованої мови ДовідникВибірка. <Ім'я>. Замість цього повсюдно використовуються запити до бази даних

Завдання 1

Здійснити підвищення швидкості проведення документа Надання послуг..

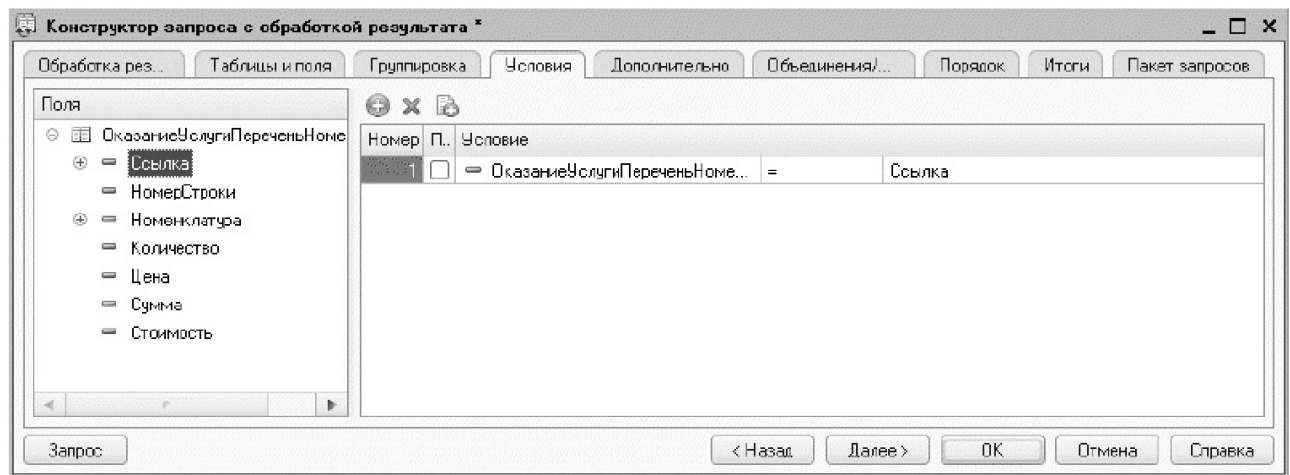
Порядок виконання

1. Відкрити модуль об'єкта конфігурації НаданняПослуг.

- Створити запит, для цього встановити курсор перед циклом обходу табличної частини документу і з контекстного меню вибрати пункт Конструктор запиту з обробкою результату. Підтвердити, що створюєте новий запит.
- В вікні конструктора запитів перейти на закладку Таблиці і поля і вибрати таблицю НаданняПослугПерелікНоменклатури – це таблична частина документу НаданняПослуг.
- З цієї таблиці Вам потрібні поля – Номенклатура, Номенклатура.ВидНоменклатури, Кількість, Сума і Вартість.
- Перейти на закладку Умови і задати умову відбору з таблиці документа тільки стрічки проведеного документа. Для цього перетягнути поле Силка в список умов запиту.

ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка

- Силка на цей документ буде передана в параметр запиту Силка.

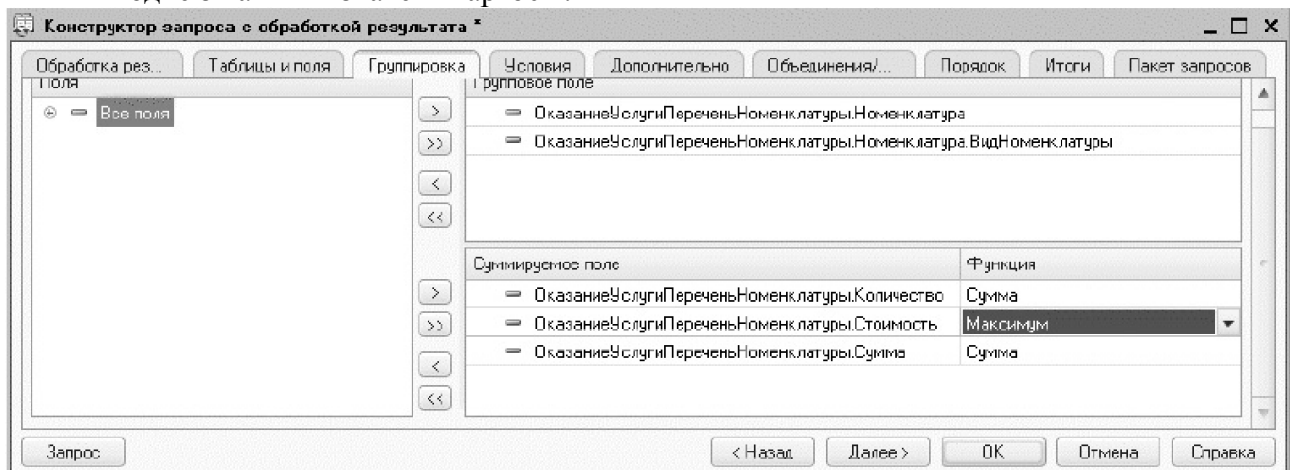


- Також потрібно врахувати, що в табличній частині документу одна і та ж номенклатура може зустрічатися декілька разів. Тому на закладці Групування згрупувати ваші запити по полю Номенклатура і Номенклатура.ВидНоменклатури, а розраховувати суму значень для полів Кількість і Сума.

Завдяки цьому в результаті значення номенклатури повторюватися не будуть, і для кожного з них будуть пораховані сумарні значення по полях Кількість і Сума, якщо в табличній частині документа міститься кілька рядків з однаковою номенклатурою.

Також до складу зсумованих полів включимо і поле Вартість. По ньому розраховувати, наприклад, функцію Максимум.

Мається на увазі, що для різних рядків однієї і тієї ж номенклатури вартість буде однаковою, тому функція Максимум потрібна нам лише для того, щоб отримати одне з наявних значень вартості.



8. На закладці Об'єднання/Псевдонім задати псевдонім для поля Кількість і Сума – КількістьВДокументі і СумаВДокументі, а для поля НоменклатураВидНоменклатура задати псевдонім ВидНоменклатури про для полегшення читання запиту.
9. Переглянути текс конструктора

```
//{{КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора внесенные вручную изменения будут
утрачены!!!

Запрос = Новый Запрос;
Запрос.Текст =
    "ВЫБРАТЬ
    |             ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
    |             ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК
    ВидНоменклатуры,
    |             СУММА (ОказаниеУслугиПереченьНоменклатуры.Количество) КАК
    КоличествоВДокументе,
    |             СУММА (ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,
    |             МАКСИМУМ (ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость
    | ИЗ
    |             Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК
    ОказаниеУслугиПереченьНоменклатуры
    | ГДЕ
    |             ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка
    |
    | СГРУППИРОВАТЬ ПО
    |             ОказаниеУслугиПереченьНоменклатуры.Номенклатура,
    |             ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";

Запрос.УстановитьПараметр ("Ссылка", Ссылка);

Результат = Запрос.Выполнить ();

ВыборкаДетальныеЗаписи = Результат.Выбрать ();

Пока ВыборкаДетальныеЗаписи.Следующий () Цикл
    // Вставить обработку выборки ВыборкаДетальныеЗаписи
КонецЦикла;

//}}КОНСТРУКТОР_ЗАПРОСА_С_ОБРАБОТКОЙ_РЕЗУЛЬТАТА
```

10. Коментарі конструктора запиту на початку та в кінці фрагменту можна видалити. Оскільки для побудови запиту було використано Конструктор запиту з обробкою результату, конструктор написав за вас код для виконання і обходу записів запиту
11. Перенести існуючий раніше в цьому модулі стрічки, які описують рух регістрів, в середину циклу обходу результату запиту

```
Пока ВыборкаДетальныеЗаписи.Следующий () Цикл
    // Вставить обработку выборки ВыборкаДетальныеЗаписи
КонецЦикла;
```

12. Для початку замість «Вставить обработку выборкиВыборкаДетальныеЗаписи» перенести умову перевірки і весь код, який формує рух по регістрах ЗалишкиМатеріалів і ВартістьМатеріалів.

Пока ВыборкаДетальныеЗаписи.Следующий () Цикл

Если ТекСтрокаПереченьНоменклатуры.Номенклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда

```
// Регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
```

```
// Регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Количество *
ТекСтрокаПереченьНоменклатуры.Стоимость;
```

КонецЕсли;

13. В умові замінити ТекСтрокаПереченьНоменклатуры.Номенклатура на
ВыборкаДетальныеЗаписи, так як вид номенклатури ви отримуєте з запиту.

14. В русі також замінити ТекСтрокаПереченьНоменклатуры на ВыборкаДетальныеЗаписи.

Пока ВыборкаДетальныеЗаписи.Следующий () Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда

```
// Регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ВыборкаДетальныеЗаписи.Количество;
```

```
// Регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
```

```
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Стоимость = ВыборкаДетальныеЗаписи.Количество *
ВыборкаДетальныеЗаписи.Стоимость;
```

КонецЕсли;

КонецЦикла;

15. Для поля Кількість ви задали псевдонім в запит, тому замінити його на
КількістьВДокументі.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
```

```
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =  
Перечисления.ВидыНоменклатуры.Материал Тогда
```

```
// Регистр ОстаткиМатериалов Расход  
Движение = Движения.ОстаткиМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Склад = Склад;
```

```
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
```

```
// Регистр СтоимостьМатериалов Расход  
Движение = Движения.СтоимостьМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе *  
ВыборкаДетальныеЗаписи.Стоимость;
```

```
КонецЕсли;
```

```
КонецЦикла;
```

16. Репер перенести формування рухів по регістру Продажі.

```
Пока ВыборкаДетальныеЗаписи.Следующий() Цикл
```

```
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =  
Перечисления.ВидыНоменклатуры.Материал Тогда
```

```
// Регистр ОстаткиМатериалов Расход  
Движение = Движения.ОстаткиМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;
```



```

Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

// Регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе *
ВыборкаДетальныеЗаписи.Стоимость;

```

КонецЕсли;

```

// Регистр Продажи
Движение = Движения.Продажи.Добавить ();
Движение.Период = Дата;
Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер = Мастер;
Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Сумма;
Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость *
ТекСтрокаПереченьНоменклатуры.Количество;

```

КонецЦикла;

17. Тут здійснимо аналогічні заміни. ТекСтрокаПереченьНоменклатуры замінити на ВыборкаДетальныеЗаписи.

Пока ВыборкаДетальныеЗаписи.Следующий () Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда

```

// Регистр ОстаткиМатериалов Расход
Движение = Движения.ОстаткиМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Склад = Склад;
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;

```

```

// Регистр СтоимостьМатериалов Расход
Движение = Движения.СтоимостьМатериалов.Добавить ();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе *
ВыборкаДетальныеЗаписи.Стоимость;

```

КонецЕсли;

```

// Регистр Продажи
Движение = Движения.Продажи.Добавить ();
Движение.Период = Дата;
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
Движение.Клиент = Клиент;
Движение.Мастер = Мастер;

```



```
Движение.Количество = ВыборкаДетальныеЗаписи.Количество;  
Движение.Выручка = ВыборкаДетальныеЗаписи.Сумма;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость *  
ВыборкаДетальныеЗаписи.Количество;
```

КонецЦикла;

18. Поля записи Сума і Кількість замінити на СумаВДокументі і КількістьВДокументі.

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

```
Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =  
Перечисления.ВидыНоменклатуры.Материал Тогда  
  
    // Регистр ОстаткиМатериалов Расход  
    Движение = Движения.ОстаткиМатериалов.Добавить();  
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
    Движение.Период = Дата;  
    Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
    Движение.Склад = Склад;  
    Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
  
    // Регистр СтоимостьМатериалов Расход  
    Движение = Движения.СтоимостьМатериалов.Добавить();  
    Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
    Движение.Период = Дата;
```

```
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе *  
ВыборкаДетальныеЗаписи.Стоимость;
```

КонецЕсли;

```
// Регистр Продажи  
Движение = Движения.Продажи.Добавить();  
Движение.Период = Дата;  
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Клиент = Клиент;  
Движение.Мастер = Мастер;  
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость *  
ВыборкаДетальныеЗаписи.КоличествоВДокументе;
```

КонецЦикла;

19. Решта тексту, який залишився можна видалити

Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл

КонецЦикла;

20. В результаті процедура проведення прийме наступний вигляд

Процедура ОбработкаПроведения(Отказ, Режим)

```
Движения.ОстаткиМатериалов.Записывать = Истина;  
Движения.СтоимостьМатериалов.Записывать = Истина;  
Движения.Продажи.Записывать = Истина;  
  
Запрос = Новый Запрос;  
Запрос.Текст =  
"ВЫБРАТЬ  
|         ОказаниеУслугиПереченьНоменклатуры.Номенклатура,  
|         ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры КАК  
ВидНоменклатуры,  
|         СУММА(ОказаниеУслугиПереченьНоменклатуры.Количество) КАК  
КоличествоВДокументе,  
|         СУММА(ОказаниеУслугиПереченьНоменклатуры.Сумма) КАК СуммаВДокументе,  
|         МАКСИМУМ(ОказаниеУслугиПереченьНоменклатуры.Стоимость) КАК Стоимость  
| ИЗ  
|         Документ.ОказаниеУслуги.ПереченьНоменклатуры КАК  
ОказаниеУслугиПереченьНоменклатуры  
| ГДЕ  
|         ОказаниеУслугиПереченьНоменклатуры.Ссылка = &Ссылка  
|  
| СГРУППИРОВАТЬ ПО  
|         ОказаниеУслугиПереченьНоменклатуры.Номенклатура,  
|         ОказаниеУслугиПереченьНоменклатуры.Номенклатура.ВидНоменклатуры";
```

```
Запрос.УстановитьПараметр("Ссылка", Ссылка);  
Результат = Запрос.Выполнить();  
ВыборкаДетальныеЗаписи = Результат.Выбрать();
```

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Материал Тогда

```
// Регистр ОстаткиМатериалов Расход  
Движение = Движения.ОстаткиМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Склад = Склад;  
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;
```

```
// Регистр СтоимостьМатериалов Расход  
Движение = Движения.СтоимостьМатериалов.Добавить();  
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;  
Движение.Период = Дата;  
Движение.Материал = ВыборкаДетальныеЗаписи.Номенклатура;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.КоличествоВДокументе *  
ВыборкаДетальныеЗаписи.Стоимость;
```

КонецЕсли;

```
// Регистр Продажи  
Движение = Движения.Продажи.Добавить();  
Движение.Период = Дата;  
Движение.Номенклатура = ВыборкаДетальныеЗаписи.Номенклатура;
```

```
Движение.Клиент = Клиент;  
Движение.Мастер = Мастер;  
Движение.Количество = ВыборкаДетальныеЗаписи.КоличествоВДокументе;  
Движение.Выручка = ВыборкаДетальныеЗаписи.СуммаВДокументе;  
Движение.Стоимость = ВыборкаДетальныеЗаписи.Стоимость *  
ВыборкаДетальныеЗаписи.КоличествоВДокументе;
```

КонецЦикла;

КонецПроцедуры

21. Запустити 1С:Підприємство в режимі отладки, перепровести документи Надання послуг і перевірити, що нічого не змінилося. **Таким чином, ви виконали перший пункт плану – позбулися в процедурі проведення від зчитування усіх даних об'єкта Номенклатура тим самим оптимізували виконання процедури проведення.**