2002 год 2 тур

Требование

Исходная ситуация. Предприятие "Витязь" торгует рубероидом одной марки. Рубероид поступает в рулонах различной длины. Учет товара ведется на счете 41, на котором предусмотрен аналитический и количественный учет по товарам. Каждый товар имеет следующие обязательные характеристики: Код, наименование, длина рулона, отпускная цена рулона.

Отпуск товара осуществляется по заказам покупателей только целыми рулонами. Количество отпускаемого товара запрашивается покупателем в метрах.

Требуется в среде программы «1С:Бухгалтерия» разработать настройку, обеспечивающую формирование счета покупателю, исходя из заказа покупателя и учетных остатков товаров.

В счет необходимо включить такое количество рулонов, чтобы их суммарная длина была не меньше запрошенной. При этом требуется, чтобы отклонение от заказанной длины было бы минимальным.

В случае если имеющийся запас товара недостаточен для удовлетворения запроса покупателя, то необходимо предусмотреть вывод соответствующего сообщения с указанием учетного остатка.

Пример

По данным учета текущие запасы товара "Рубероид РБ22" составляют:

|  |  |  |
| --- | --- | --- |
| Метраж рулона | Кол-во рулонов | Отпускная цена |
| 27 | 12 | 270 |
| 45 | 11 | 450 |
| 33 | 21 | 330 |
| 60 | 0 | 600 |
| 22 | 17 | 220 |
| 47 | 9 | 470 |

Покупатель ООО "Фабрика грез" затребовал 260 м. товара "Рубероид РБ22".   
На основании данного требования программа сформировала следующий счет.

Счет № от " " 2000 г.

Покупатель: ООО "Фабрика грез"

Заказано: 260 м

К оплате:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Метраж рулона | Кол-во рулонов | Метров | Сумма |
| Рубероид РБ22 | 27 | 8 | 216 | 2160 |
| Рубероид РБ22 | 22 | 2 | 44 | 440 |
| Итого |  |  | 260 | 2600 |

Решение

Для решения данной задачи требуется рассмотреть возможные варианты комбинирования рулонов в группы, подходящие под условие задачи. Используем рекурсивную функцию, для обхода всего графа комбинаций. Условием остановки поиска будет равенство длинны рулонов требуемой длине, если же варианта с равенством нет, то выбираем с минимальным отклонением от цели.

Реализация

&НаКлиенте

Процедура Сформировать(Команда)

СформироватьНаСервере();

КонецПроцедуры

&НаСервере

Процедура СформироватьНаСервере()

ЭтотОбъект=РеквизитФормыВЗначение("Отчет");

Макет = ЭтотОбъект.ПолучитьМакет("Макет");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналПроводокОстатки.Субконто1 КАК Рулон,

| ЖурналПроводокОстатки.Субконто2 КАК Длина,

| ЖурналПроводокОстатки.Субконто3 КАК Цена,

| ЖурналПроводокОстатки.КоличествоОстаток КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки КАК ЖурналПроводокОстатки

|ГДЕ

| ЖурналПроводокОстатки.Счет = &Счет

| И ЖурналПроводокОстатки.Субконто1.Ссылка = &Рулон

|

|УПОРЯДОЧИТЬ ПО

| Длина";

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Товары);

Запрос.УстановитьПараметр("Рулон", Рулон);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Остатки = Новый Массив;

Пока Выборка.Следующий() Цикл

Данные = Новый Структура;

Данные.Вставить("Длина", Выборка.Длина);

Данные.Вставить("Цена", Выборка.Цена);

Данные.Вставить("Количество", Выборка.Количество);

Остатки.Добавить(Данные);

КонецЦикла;

Результат = НайтиЛучшийПуть(Остатки, Неопределено, Длина);

ТабДок.Очистить();

Рулоны = Новый Массив;

Для Каждого Элемент Из Результат.Путь Цикл

Найден = Ложь;

Для Каждого РулонЭлемент Из Рулоны Цикл

Если РулонЭлемент.Длина = Элемент.Длина Тогда

РулонЭлемент.Количество = РулонЭлемент.Количество + 1;

РулонЭлемент.Сумма = РулонЭлемент.Сумма + Элемент.Цена;

Найден = Истина;

Прервать;

КонецЕсли;

КонецЦикла;

Если Найден = Ложь Тогда

РулонЭлемент = Новый Структура;

РулонЭлемент.Вставить("Длина", Элемент.Длина);

РулонЭлемент.Вставить("Количество", 1);

РулонЭлемент.Вставить("Сумма", Элемент.Цена);

Рулоны.Добавить(РулонЭлемент);

КонецЕсли;

КонецЦикла;

ЗаголовокОбласть = Макет.ПолучитьОбласть("Заголовок");

ДанныеОбласть = Макет.ПолучитьОбласть("Данные");

ИтогОбласть = Макет.ПолучитьОбласть("Итог");

ТабДок.Вывести(ЗаголовокОбласть);

Метров = 0;

Сумма = 0;

Для Каждого РулонЭлемент Из Рулоны Цикл

ДанныеОбласть.Параметры.Рулон = Рулон;

ДанныеОбласть.Параметры.Метраж = РулонЭлемент.Длина;

ДанныеОбласть.Параметры.Количество = РулонЭлемент.Количество;

ДанныеОбласть.Параметры.Метров = РулонЭлемент.Длина \* РулонЭлемент.Количество;

ДанныеОбласть.Параметры.Сумма = РулонЭлемент.Сумма;

ТабДок.Вывести(ДанныеОбласть);

Метров = Метров + ДанныеОбласть.Параметры.Метров;

Сумма = Сумма + ДанныеОбласть.Параметры.Сумма;

КонецЦикла;

ИтогОбласть.Параметры.Метров = Метров;

ИтогОбласть.Параметры.Сумма = Сумма;

ТабДок.Вывести(ИтогОбласть);

КонецПроцедуры

&НаСервере

Функция НайтиЛучшийПуть(Остатки, ПрошлыйРезультат, Цель)

Если ПрошлыйРезультат = Неопределено Тогда

ПрошлыйРезультат = Новый Структура;

ПрошлыйРезультат.Вставить("Длина", 0);

ПрошлыйРезультат.Вставить("Путь", Новый Массив);

КонецЕсли;

ЛучшийПуть = Неопределено;

ЛучшийЭлемент = Неопределено;

ЛучшаяДлина = 0;

Для Каждого Остаток Из Остатки Цикл

Если Остаток.Количество = 0 Тогда

Продолжить;

КонецЕсли;

НоваяДлина = ПрошлыйРезультат.Длина + Остаток.Длина;

Если НоваяДлина < Цель Тогда

Остаток.Количество = Остаток.Количество - 1;

ПрошлыйРезультат.Длина = НоваяДлина;

ПрошлыйРезультат.Путь.Добавить(Остаток);

НовыйРезультат = НайтиЛучшийПуть(Остатки, ПрошлыйРезультат, Цель);

ПрошлыйРезультат.Путь.Удалить(ПрошлыйРезультат.Путь.ВГраница());

ПрошлыйРезультат.Длина = ПрошлыйРезультат.Длина - Остаток.Длина;

Остаток.Количество = Остаток.Количество + 1;

Если НовыйРезультат = Неопределено Тогда

Продолжить;

КонецЕсли;

Если НовыйРезультат.Длина >= Цель Тогда

Если ЛучшаяДлина = 0 ИЛИ (ЛучшаяДлина - Цель) > (НовыйРезультат.Длина - Цель) Тогда

ЛучшаяДлина = НовыйРезультат.Длина;

ЛучшийПуть = НовыйРезультат.Путь;

ЛучшийЭлемент = Неопределено;

КонецЕсли;

КонецЕсли;

Иначе

Если ЛучшаяДлина = 0 ИЛИ (ЛучшаяДлина - Цель) > (НоваяДлина - Цель) Тогда

ЛучшаяДлина = НоваяДлина;

ЛучшийЭлемент = Остаток;

ЛучшийПуть = Неопределено;

КонецЕсли;

КонецЕсли;

Если ЛучшаяДлина = Цель Тогда

Прервать;

КонецЕсли;

КонецЦикла;

Если ЛучшийЭлемент <> Неопределено Тогда

ЛучшийПуть = Новый Массив;

Для Каждого Прошлый Из ПрошлыйРезультат.Путь Цикл

ЛучшийПуть.Добавить(Прошлый);

КонецЦикла;

ЛучшийПуть.Добавить(ЛучшийЭлемент);

КонецЕсли;

Если ЛучшийПуть = Неопределено Тогда

Возврат Неопределено;

КонецЕсли;

Результат = Новый Структура;

Результат.Вставить("Длина", ЛучшаяДлина);

Результат.Вставить("Путь", ЛучшийПуть);

Возврат Результат;

КонецФункции

Автор: Михайлов Алексей

Пояснения к коду:

&НаСервере

Процедура СформироватьНаСервере()

ЭтотОбъект=РеквизитФормыВЗначение("Отчет");

Макет = ЭтотОбъект.ПолучитьМакет("Макет");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналПроводокОстатки.Субконто1 КАК Рулон,

| ЖурналПроводокОстатки.Субконто2 КАК Длина,

| ЖурналПроводокОстатки.Субконто3 КАК Цена,

| ЖурналПроводокОстатки.КоличествоОстаток КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки КАК ЖурналПроводокОстатки

|ГДЕ

| ЖурналПроводокОстатки.Счет = &Счет

| И ЖурналПроводокОстатки.Субконто1.Ссылка = &Рулон

|

|УПОРЯДОЧИТЬ ПО

| Длина";

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Товары);

Запрос.УстановитьПараметр("Рулон", Рулон);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Остатки = Новый Массив;

Пока Выборка.Следующий() Цикл

Данные = Новый Структура;

Данные.Вставить("Длина", Выборка.Длина);

Данные.Вставить("Цена", Выборка.Цена);

Данные.Вставить("Количество", Выборка.Количество);

Остатки.Добавить(Данные);

КонецЦикла;

Результат = НайтиЛучшийПуть(Остатки, Неопределено, Длина);

ТабДок.Очистить();

Рулоны = Новый Массив;

Для Каждого Элемент Из Результат.Путь Цикл

Найден = Ложь;

Для Каждого РулонЭлемент Из Рулоны Цикл

Если РулонЭлемент.Длина = Элемент.Длина Тогда

РулонЭлемент.Количество = РулонЭлемент.Количество + 1;

РулонЭлемент.Сумма = РулонЭлемент.Сумма + Элемент.Цена;

Найден = Истина;

Прервать;

КонецЕсли;

КонецЦикла;

Если Найден = Ложь Тогда

РулонЭлемент = Новый Структура;

РулонЭлемент.Вставить("Длина", Элемент.Длина);

РулонЭлемент.Вставить("Количество", 1);

РулонЭлемент.Вставить("Сумма", Элемент.Цена);

Рулоны.Добавить(РулонЭлемент);

КонецЕсли;

КонецЦикла;

ЗаголовокОбласть = Макет.ПолучитьОбласть("Заголовок");

ДанныеОбласть = Макет.ПолучитьОбласть("Данные");

ИтогОбласть = Макет.ПолучитьОбласть("Итог");

ТабДок.Вывести(ЗаголовокОбласть);

Метров = 0;

Сумма = 0;

Для Каждого РулонЭлемент Из Рулоны Цикл

ДанныеОбласть.Параметры.Рулон = Рулон;

ДанныеОбласть.Параметры.Метраж = РулонЭлемент.Длина;

ДанныеОбласть.Параметры.Количество = РулонЭлемент.Количество;

ДанныеОбласть.Параметры.Метров = РулонЭлемент.Длина \* РулонЭлемент.Количество;

ДанныеОбласть.Параметры.Сумма = РулонЭлемент.Сумма;

ТабДок.Вывести(ДанныеОбласть);

Метров = Метров + ДанныеОбласть.Параметры.Метров;

Сумма = Сумма + ДанныеОбласть.Параметры.Сумма;

КонецЦикла;

ИтогОбласть.Параметры.Метров = Метров;

ИтогОбласть.Параметры.Сумма = Сумма;

ТабДок.Вывести(ИтогОбласть);

КонецПроцедуры

Данная процедура – обработчик нажатия кнопки Сформировать на сервере, в ней происходит получение данных об остатках рулонов в базе, составление массива остатков, вызов рекурсивной функции поиска пути, подсчет количества рулонов с группировкой по длине и вывод результата в табличный документ.

ЭтотОбъект=РеквизитФормыВЗначение("Отчет");

Макет = ЭтотОбъект.ПолучитьМакет("Макет");

В данном блоке производится получение макета, так как отчет – внешний, то чтобы получить его макет требуется перевести реквизит формы Отчет в объект и получить его макет.

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЖурналПроводокОстатки.Субконто1 КАК Рулон,

| ЖурналПроводокОстатки.Субконто2 КАК Длина,

| ЖурналПроводокОстатки.Субконто3 КАК Цена,

| ЖурналПроводокОстатки.КоличествоОстаток КАК Количество

|ИЗ

| РегистрБухгалтерии.ЖурналПроводок.Остатки КАК ЖурналПроводокОстатки

|ГДЕ

| ЖурналПроводокОстатки.Счет = &Счет

| И ЖурналПроводокОстатки.Субконто1.Ссылка = &Рулон

|

|УПОРЯДОЧИТЬ ПО

| Длина";

Запрос.УстановитьПараметр("Счет", ПланыСчетов.ПланСчетов.Товары);

Запрос.УстановитьПараметр("Рулон", Рулон);

Результат = Запрос.Выполнить();

Выборка = Результат.Выбрать();

Остатки = Новый Массив;

Пока Выборка.Следующий() Цикл

Данные = Новый Структура;

Данные.Вставить("Длина", Выборка.Длина);

Данные.Вставить("Цена", Выборка.Цена);

Данные.Вставить("Количество", Выборка.Количество);

Остатки.Добавить(Данные);

КонецЦикла;

В данном блоке производится получение остатков указанных рулонов на счете Товары и запись результатов в массив Остатки, записи которого являются структурами с полями Длина, Цена, Количество.

Результат = НайтиЛучшийПуть(Остатки, Неопределено, Длина);

Данная строка вызывает рекурсивную функцию поиска пути, первый параметр – начальные остатки рулонов, второй параметр – предыдущие выбранные рулоны (на начало поиска их нет, поэтому Неопределено) и третий параметр – требуемая длина. Возвращает функция структуру с полями Длина и Путь (массив из структур аналогичный массиву Остатки).

Рулоны = Новый Массив;

Для Каждого Элемент Из Результат.Путь Цикл

Найден = Ложь;

Для Каждого РулонЭлемент Из Рулоны Цикл

Если РулонЭлемент.Длина = Элемент.Длина Тогда

РулонЭлемент.Количество = РулонЭлемент.Количество + 1;

РулонЭлемент.Сумма = РулонЭлемент.Сумма + Элемент.Цена;

Найден = Истина;

Прервать;

КонецЕсли;

КонецЦикла;

Если Найден = Ложь Тогда

РулонЭлемент = Новый Структура;

РулонЭлемент.Вставить("Длина", Элемент.Длина);

РулонЭлемент.Вставить("Количество", 1);

РулонЭлемент.Вставить("Сумма", Элемент.Цена);

Рулоны.Добавить(РулонЭлемент);

КонецЕсли;

КонецЦикла;

В данном блоке полученный результат разбивается на группы рулонов, по полю Длина. Для каждой записи из Результат.Путь ищется в массиве Рулоны элемент, с такой же длиной, если элемент найден то у него добавляется количество и меняется сумма, если не найден то элемент создается и добавляется в массив Рулоны.

ЗаголовокОбласть = Макет.ПолучитьОбласть("Заголовок");

ДанныеОбласть = Макет.ПолучитьОбласть("Данные");

ИтогОбласть = Макет.ПолучитьОбласть("Итог");

ТабДок.Вывести(ЗаголовокОбласть);

Метров = 0;

Сумма = 0;

Для Каждого РулонЭлемент Из Рулоны Цикл

ДанныеОбласть.Параметры.Рулон = Рулон;

ДанныеОбласть.Параметры.Метраж = РулонЭлемент.Длина;

ДанныеОбласть.Параметры.Количество = РулонЭлемент.Количество;

ДанныеОбласть.Параметры.Метров = РулонЭлемент.Длина \* РулонЭлемент.Количество;

ДанныеОбласть.Параметры.Сумма = РулонЭлемент.Сумма;

ТабДок.Вывести(ДанныеОбласть);

Метров = Метров + ДанныеОбласть.Параметры.Метров;

Сумма = Сумма + ДанныеОбласть.Параметры.Сумма;

КонецЦикла;

ИтогОбласть.Параметры.Метров = Метров;

ИтогОбласть.Параметры.Сумма = Сумма;

ТабДок.Вывести(ИтогОбласть);

В данном блоке производится получение областей макета, вывод заголовка, вывод каждого элемента массива Рулоны с суммированием метров и суммы, и вывод итогов.

&НаСервере

Функция НайтиЛучшийПуть(Остатки, ПрошлыйРезультат, Цель)

Если ПрошлыйРезультат = Неопределено Тогда

ПрошлыйРезультат = Новый Структура;

ПрошлыйРезультат.Вставить("Длина", 0);

ПрошлыйРезультат.Вставить("Путь", Новый Массив);

КонецЕсли;

ЛучшийПуть = Неопределено;

ЛучшийЭлемент = Неопределено;

ЛучшаяДлина = 0;

Для Каждого Остаток Из Остатки Цикл

Если Остаток.Количество = 0 Тогда

Продолжить;

КонецЕсли;

НоваяДлина = ПрошлыйРезультат.Длина + Остаток.Длина;

Если НоваяДлина < Цель Тогда

Остаток.Количество = Остаток.Количество - 1;

ПрошлыйРезультат.Длина = НоваяДлина;

ПрошлыйРезультат.Путь.Добавить(Остаток);

НовыйРезультат = НайтиЛучшийПуть(Остатки, ПрошлыйРезультат, Цель);

ПрошлыйРезультат.Путь.Удалить(ПрошлыйРезультат.Путь.ВГраница());

ПрошлыйРезультат.Длина = ПрошлыйРезультат.Длина - Остаток.Длина;

Остаток.Количество = Остаток.Количество + 1;

Если НовыйРезультат = Неопределено Тогда

Продолжить;

КонецЕсли;

Если НовыйРезультат.Длина >= Цель Тогда

Если ЛучшаяДлина = 0 ИЛИ (ЛучшаяДлина - Цель) > (НовыйРезультат.Длина - Цель) Тогда

ЛучшаяДлина = НовыйРезультат.Длина;

ЛучшийПуть = НовыйРезультат.Путь;

ЛучшийЭлемент = Неопределено;

КонецЕсли;

КонецЕсли;

Иначе

Если ЛучшаяДлина = 0 ИЛИ (ЛучшаяДлина - Цель) > (НоваяДлина - Цель) Тогда

ЛучшаяДлина = НоваяДлина;

ЛучшийЭлемент = Остаток;

ЛучшийПуть = Неопределено;

КонецЕсли;

КонецЕсли;

Если ЛучшаяДлина = Цель Тогда

Прервать;

КонецЕсли;

КонецЦикла;

Если ЛучшийЭлемент <> Неопределено Тогда

ЛучшийПуть = Новый Массив;

Для Каждого Прошлый Из ПрошлыйРезультат.Путь Цикл

ЛучшийПуть.Добавить(Прошлый);

КонецЦикла;

ЛучшийПуть.Добавить(ЛучшийЭлемент);

КонецЕсли;

Если ЛучшийПуть = Неопределено Тогда

Возврат Неопределено;

КонецЕсли;

Результат = Новый Структура;

Результат.Вставить("Длина", ЛучшаяДлина);

Результат.Вставить("Путь", ЛучшийПуть);

Возврат Результат;

КонецФункции

Данная функция рекурсивна, производит поиск такой комбинации остатков рулонов, которая в длине будет не меньше указанной. Для начала проверяется есть ли предыдущий результат, то есть вызвана ли функция рекурсивно. Если предыдущий результат не определен, то структура создается, с начальными данными (путь пуст и длина 0). Далее происходит проверка каждого рулона из остатков, количество которого не нулевое. Проверяется следующим образом – длина рулона прибавляется к начальной (предыдущий результат), если новая длина меньше требуемой, то рекурсивно вызываем функцию, предварительно уменьшив количество выбранных рулонов и указав нужный предыдущий результат, после получения рекурсивного результата отменяем изменения, чтобы продолжить поиск по другим веткам графа. Если длина по данной ветке больше либо равна целевой и лучше, чем предыдущий сохраненный результат или такого нет, то запоминаем данный выбор. Если же новая длина больше либо равна целевой, то опять же сверяемся с предыдущим сохраненным результатом. Далее проверяем насколько точно подходит результат, если длина точно равна целевой – завершаем поиск. После обхода всех рулонов, составляется структура результата с полями Длина и Путь.

Сформированный отчет

