

УДК 004.421: 339.5

АЛГОРИТМЫ ПРЕОБРАЗОВАНИЯ И СРАВНЕНИЯ СПИСКОВ ДАННЫХ, ПОЛУЧЕННЫХ ИЗ ТАМОЖЕННЫХ ДЕКЛАРАЦИЙ

Сальников В.И.

Санкт-Петербургский имени В.Б.Бобкова филиал Российской таможенной академии

ALGORITHMS FOR TRANSFORMING AND COMPARING DATA LISTS OBTAINED FROM CUSTOMS DECLARATIONS

Salnikov V.I.

St. Petersburg named after V.B. Bobkov branch of the Russian Customs Academy

Аннотация

В статье представлены алгоритмы, разработанные для решения задачи сравнительного анализа списков данных, полученных из таможенных баз данных. Алгоритмы позволяют выполнить сравнение списков данных, привести данные к единому виду, осуществить поиск различий в данных и вывести комментарии о характере различий.

В статье выполнен анализ различных ситуаций, возникающих в процессе обработки данных. Показана упрощенная структура сравниваемых списков. Для построения алгоритмов предложено конвертировать сравниваемые списки в таблицы приложения Microsoft Access и построить различные запросы к ним в зависимости от ситуации. Показана структура запросов, построенных с использованием языка SQL. Представлены результаты выполнения запросов.

Разработанные алгоритмы способны существенно снизить трудоемкость и ускорить работу таможенных инспекторов, в чьи обязанности входит сравнение больших списков данных, повысить эффективность их деятельности в процессе выявления фактов занижения таможенной стоимости товаров.

Ключевые слова: алгоритм, таможенное дело, таможенное декларирование, таможенная стоимость, анализ данных, база данных, запросы, SQL.

Одной из проблем, возникающих в процессе декларирования товаров, перемещаемых через таможенную территорию ЕАЭС, является занижение таможенной стоимости товаров. Это позволяет участникам внешнеэкономической деятельности снизить величину таможенных платежей, налога на добавленную стоимость и акцизов [1], что, в свою очередь, снижает поступление денежных средств в государственный бюджет. Для выявления нарушений таможенного законодательства данного вида и борьбы с ними таможенным инспекторам отделов

Abstract

The article presents algorithms designed to solve the problem of comparative analysis of data lists obtained from customs databases. Algorithms allow you to compare data lists, bring the data to a single form, search for differences in the data and display comments on the nature of the differences.

The article analyzes various situations that arise during data processing. The simplified structure of the compared lists is shown. To build algorithms, it is proposed to convert the compared lists into tables of the Microsoft Access application and build various queries for them, depending on the situation. The structure of queries constructed using the SQL language is shown. The results of query execution are presented.

The developed algorithms can significantly reduce the complexity and speed up the work of customs inspectors, whose responsibility is to compare large lists of data, increase the efficiency of their activities in the process of identifying facts of understating the customs value of goods.

Keywords: algorithm, customs affairs, customs declaration, customs value, data analysis, database, queries, SQL.

контроля таможенной стоимости в процессе выполнения служебных обязанностей приходится решать разнообразные аналитические задачи, связанные с выборкой данных из баз данных, в которых хранятся сведения из таможенных деклараций, сопоставлению различных выборок и списков данных с целью выявления наличия и оценки величины расхождений задекларированных значений с нормативными.

Особенность данного вида аналитических задач заключается в высокой трудоем-

кости их решения, связанной не только с использованием информационных технологий и программных средств, но и с монотонным, рутинным характером вычислительных и логических операций, выполняемых таможенным инспектором. Актуальность темы настоящего исследования в свете повышения требований к эффективности деятельности таможенных органов определяется тем, что решение задачи автоматизации выполнения рутинных операций в таможенных позволяет значительно ускорить процесс решения аналитических задач и существенно повысить качество принимаемых управленческих решений.

Для осуществления автоматизации решения аналитических задач необходима разработка соответствующих алгоритмов, которые могут быть реализованы приложениях, доступных таможенным инспекторам на рабочих местах.

Универсальным инструментом, который применяется для построения алгоритмов решения аналитических задач, является язык запросов SQL, поскольку его инструкции могут быть использованы для взаимодействия с различными системами управления базами данных. Предлагается строить алгоритмы решения задач в форме запросов с использованием графического интерфейса приложения Microsoft Access, а затем конвертировать их средствами данного приложения в инструкции языка SQL. В случае если анализируемые данные содержатся в разных источниках, то предлагается импортировать их в таблицы Microsoft Access, а затем выполнить запрос на сопоставление данных. Запрос необходимо построить таким образом, чтобы он не только выбирал записи со значимыми различиями, но и выводил комментарий по сути каждого различия.

В предыдущих работах автором уже предпринималась попытка построения подобных алгоритмов. К примеру, в работе [2] был разработан алгоритм отбора таможенных деклараций, вместе с которыми предоставлены ненадлежащие документы или предоставлены не все необходимые документы. В работе [3] автором был разработан алгоритм определения вида ставки таможенной пошлины в зависимости от страны происхождения товара.

В данной работе автором ставится задача построения алгоритмов приведения

списков данных к единому формату и сравнению их с целью поиска значимых различий.

Для решения задачи рассмотрим три ситуации, которые могут возникнуть в процессе обработки данных.

Ситуация 1. Имеются две таблицы данных по таможенному декларированию, в одной из которых хранятся нормативные значения некоторых показателей, а во второй – данные, позволяющие рассчитать фактические значения показателей. Требуется из второй таблицы выбрать такие записи, в которых фактические значения показателей отличаются в меньшую сторону от нормативных значений на заданную величину.

В качестве примера рассмотрим две связанные таблицы. В таблице «Декларации» представлены условные данные о таможенной стоимости партии товара и количестве товара в ней. В таблице «Справочник средних цен» представлены средние (нормативные) цены на товар каждого вида. Таблицы связаны между собой по полю «Товар».

Требуется определить товары, цены на которые занижены по сравнению со средней ценой более чем на 30%. Упрощенные модели таблиц представлены на рис. 1 и рис. 2.

Для решения поставленной задачи предлагается создать запрос на основе двух указанных таблиц, назовем его «Проверка уровня цен». Запрос должен отображать данные о товаре и вычисляемое поле, в котором рассчитывается фактическая цена товара по формуле:

Цена_факт: [Там_стоимость]/[количество]

Кроме того, в данное вычисляемое поле необходимо записать условие отбора: <[Цена_сред]*0,7 (здесь для записи формул используются нотации, принятые в Microsoft Access).

Таким образом, для того, чтобы запись попала в выборку, фактическая цена должна быть меньше 70% от средней (нормативной) цены.

Запись запроса на языке SQL имеет следующий вид:

SELECT Декларации.[№_декл], Декларации.Товар,

Декларации.Там_стоимость, Декларации.количество,

[Там_стоимость]/[количество]

AS Цена_факт, [Справочник средних цен].Цена_сред

Код	№_декл	Товар	Там_стоимость	количество
1	1111	А	120 000 Р	600
2	1111	Б	255 000 Р	850
3	1112	А	144 000 Р	1200
4	1112	В	1 975 000 Р	3950
5	1113	Г	180 000 Р	300
6	1121	А	49 000 Р	490
7	1121	Д	235 200 Р	980
8	1122	Д	201 000 Р	1340

Рис. 1. Пример таблицы «Декларации»

Товар	Цена_сред
А	150 Р
Б	340 Р
В	430 Р
Г	610 Р
Д	220 Р

Рис. 2. Пример таблицы «Справочник средних цен»

```

FROM Декларации
INNER JOIN [Справочник средних цен]
ON Декларации.Товар = [Справочник
средних цен].Товар
WHERE ((([Там_стоимость]/[количе-
ство])<[Цена_сред]*0.7))

```

Результат выполнения запроса показан на рис. 3. В запросе автоматически отобраны записи о двух товарах, цена на которые существенно занижена.

Ситуация 2. Имеются две таблицы данных по таможенному декларированию, в одной из которых хранятся данные, представленные в одном формате, а во второй – данные, представленные, возможно, в другом формате. Требуется предварительно привести данные к единому формату, например, используя строковые функции, а

затем выбрать записи, которые в двух списках не идентичны. В случае попадания записи в выборку необходимо вывести комментарий о причинах такого попадания. Предполагается, что списки данных имеют ключевые поля и не содержат пропусков в них.

В качестве примера рассмотрим две несвязанные таблицы, при этом предполагается, что последовательность записей в таблицах одинакова. Поле «№_строки» в каждой таблице является ключевым полем, поэтому записи с одинаковым номером в разных таблицах – это на самом деле информация об одной и той же декларации и одном и том же товаре в ее рамках.

№_декл	Товар	Там_стоимость	количество	Цена_факт	Цена_сред
1121	А	49 000 Р	490	100 Р	150 Р
1122	Д	201 000 Р	1340	150 Р	220 Р

Рис. 3. Результат выборки товаров с заниженными ценами

Принимая во внимание данное обстоятельство, для решения задачи предлагается установить связь между таблицами непосредственно в запросе. Примеры таблиц приведены на рис. 4 и рис. 5.

Анализ таблиц позволяет сделать вывод о том, что в первой таблице «Список1» номер декларации в поле «ТД» сформирован из кода таможенного органа, даты в шестизначном формате и семизначного номера декларации, разделенных знаком «/».

Во второй таблице «Список2» вместо данного поля присутствуют три поля: «КодТО», «Дата», «Номер_ТД». Дата представлена с разделителями-точками, а номер декларации – без нулей. Следовательно, для приведения списков к единообразному виду, чтобы получить полный номер декларации на товары, необходимо во второй таблице объединить указанные

поля, одновременно убрав точки в дате и добавив недостающие нули в номере декларации. С данной целью предлагается создать запрос, содержащий вычисляемое поле «ТД2», используя формулу:

ТД2: [КодТО] & "/" & Left([Дата];2) & Mid([Дата];4;2) & Right([Дата];2) & "/" & Right("0000000" & [Номер_ТД];7).

В формуле используется оператор объединения строк, а также строковые функции, необходимые для преобразования даты с удалением точек и для добавления недостающих нулей перед номером декларации.

Аналогично в таблице «Список2» в поле «Код_товара» в некоторых записях утерян ноль – начало кода товара. Для

№_строки	ТД	Код_товара	Количество
1	10216000/011219/0000001	0901210009	2000
2	10216000/021219/0000002	0902300009	3500
3	10216000/150120/0000017	0803901000	40000
4	10009190/160120/0000021	2204101100	2500
5	10009190/170120/0000022	8711100000	120
6	10210000/171219/0000024	8516108000	300

Рис. 4. Таблица, содержащая первый список

№_строки	КодТО	Дата	Номер_ТД	Код_товара	Количество2
2	10216000	02.12.2019	2	902300009	2000
3	10216000	15.01.2020	17	803901000	40000
4	10009190	16.01.2020	21	2204101100	2500
5	10009190	17.01.2020	22	8712003000	120
6	10210000	17.01.2020	23	8516108000	300
1	10216000	01.12.2019	1	901210009	2000

Рис. 5. Таблица, содержащая второй список

устранения ошибки в коде товара предлагается использовать в запросе вычисляемое поле:

Код_товара2: Right("0" & Список2.[Код_товара],10)

Помимо вычисляемых полей, в запросе должны присутствовать остальные поля обеих таблиц, по которым производится сопоставление данных. Далее необходимо записать условия отбора для полей, по которым производится сравнение данных:

- для поля «ТД2» записывается < > [ТД];
- для поля «Количество2» записывается < > [Количество];
- для поля «Код_товара2» записывается < > [Код_товара].

Такая запись означает, что для того, чтобы данные попали в выборку, необходимо, чтобы данные из полей одной таблицы не были равны данным из полей другой таблицы хотя бы по одному полю. Следовательно, в данном случае необходимо использовать логический оператор Or, соответственно условия отбора записываются в разных строках конструктора запросов:

- первое условие – в строке «Условие отбора»;
- второе – в строке «Или»;
- третье – в строке, следующей за строкой «Или» и т.д.

Далее в запросе необходимо создать вычисляемое поле, характеризующее вид несоответствия, найденного в результате работы запроса. Это может быть несоответствие номера декларации, несоответствие кода товара и несоответствие количества. Аналогично можно добавить и другие характеристики. Для создания такого поля пред-

лагается использовать формулу, содержащую встроенную функцию множественного выбора Switch:

Вид_ошибки: Switch([ТД2]< >[ТД]; "Неправильный номер ТД";

[Код_товара2]< >Список1.[Код_товара]; "Неправильный код товара"; [Количество2]< >[Количество];"Неправильное количество")

На языке SQL данный запрос имеет вид:

```
SELECT Список1.ТД, Список1.Код_товара,
Список1.Количество, КодТО & "/" &
Left(Дата,2) & Mid(Дата,4,2) & Right(Дата,2)
& "/" & Right("0000000" & Номер_ТД,7)
```

```
AS ТД2, Right("0" & Список2.Код_товара,10)
```

```
AS Код_товара2, Список2.Количество2,
Switch(ТД2< >ТД, "Неправильный номер ТД",
```

```
Код_товара2< >Список1.Код_товара, "Неправильный код товара",
Количество2< >Количество, "Неправильное количество")
```

```
AS Вид_ошибки
FROM Список1 INNER JOIN Список2
ON Список1.[№_строки] = Список2.[№_строки]
```

```
WHERE КодТО & "/" & Left(Дата,2) &
Mid(Дата,4,2) &
```

```
Right(Дата,2) & "/" & Right("0000000" &
Номер_ТД,7)< >ТД
```

```
OR Right("0" & Список2.Код_товара,10)< >Список1.Код_товара
```

```
OR Список2.Количество2< >Количество
```

При построении SQL-запроса некоторые поля могут относиться к нескольким таблицам или запросам. В данном случае

ТД	Код_товара	Количество	ТД2	Код_товара2	Количество2	Вид_ошибки
10216000/021219/0000002	0902300009	3500	10216000/021219/0000002	0902300009	2000	Неправильное количество
10009190/170120/0000022	8711100000	120	10009190/170120/0000022	8712003000	120	Неправильный код товара
10210000/171219/0000024	8516108000	300	10210000/170120/0000023	8516108000	300	Неправильный номер ТД

Рис. 6. Результат выполнения запроса на сравнение при наличии ключевых полей

для таких полей дополнительно указывается содержащая их таблица. Для остальных полей такая адресация необязательна. Также в SQL-запросе можно опустить квадратные скобки в именах полей. В некоторых случаях такие скобки могут вставляться автоматически при переходе в режим конструктора запросов.

В результате выполнения запроса получаем таблицу, показанную на рис. 6. В ней отражены декларации, имеющие расхождения в данных, полученные путем сопоставления двух списков. В поле «Вид_ошибки» сформирован соответствующий комментарий. Выявлены три декларации с ошибками, и приведены три различных комментария, что подтверждает корректную работу формулы.

Ситуация 3. Требуется решить аналогичную задачу, осложненную тем, что в исходных данных в одной или обеих таблицах отсутствует поле «Номер строки». В данном случае ключевое поле (поля) предлагается создать искусственно, объединив данные нескольких известных полей так, чтобы данные во вновь создаваемом поле стали уникальными.

В представленном примере ключевыми полями будут считаться поля «ТД» и «ТД2», отображающие номер декларации. Поле «ТД2» формируется так же, как и во втором случае. В реальной ситуации, если предложенных полей не хватает для однозначной идентификации записи, то возможно присоединение дополнительной информации к полям, отображающим номер декларации, например, кода товара и других данных.

Для корректности формирования выборки необходимо в запросе установить связь между полями, которые указаны как ключевые. Однако возможна ситуация, когда списки данных не совпадают, и данные о декларациях, присутствующие в первом списке, отсутствуют во втором списке и

наоборот. Такие данные также необходимо найти. Но если в одной из сравниваемых таблиц имеется запись с таким значением ключевого поля, которое отсутствует в ключевом поле другой таблицы, то такая запись в запросе не отобразится.

Для устранения отмеченного недостатка возможны два способа решения задачи. Первый способ предполагает создание вспомогательной таблицы «Список3» для преобразования данных, а затем построение запроса на объединение двух вложенных запросов на выборку. Оба вложенных запроса строятся на основе двух таблиц «Список1» и «Список3», связь между которыми устанавливается с помощью операторов языка SQL LEFT JOIN и RIGHT JOIN. Данные операторы позволяют выбрать все записи из одной таблицы и такие записи из другой таблицы, которые совпадают с данными из первой таблицы по связанному полю. Поэтому комбинация запросов с указанными операторами позволит получить все записи из обеих таблиц без потерь данных.

Для реализации данного алгоритма сначала требуется создать таблицу «Список3», которая получается путем преобразования данных таблицы «Список2». Инструкция SQL для создания таблицы с помощью запроса выглядит следующим образом:

```
SELECT [КодТО] & "/" & Left([Дата],2) &
Mid([Дата],4,2) &
Right([Дата],2) & "/" & Right("0000000" &
[Номер_ТД],7)
AS ТД2, Right("0" & [Список2].[Код_товара],10)
AS Код_товара2, Список2.Количество2
INTO Список3
FROM Список2
```

Далее на основе таблиц «Список1» и «Список3» строится запрос на объединение двух выборок. Поскольку при построении

выборки с использованием операторов LEFT JOIN и RIGHT JOIN возникают пустые ячейки, то при формировании условия отбора и комментария используется сравнение Is Null. Остальные условия отбора используются такие же, как и в предыдущем алгоритме. Запрос в режиме SQL представлен далее:

```
SELECT Список1.ТД, Список1.Код_товара,
Список1.Количество,
Список3.ТД2, Список3.Код_товара2,
Список3.Количество2,
Switch(ТД2 Is Null OR ТД Is Null, "Неправильный номер ТД",
Код_товара2 < > Код_товара, "Неправильный код товара",
Количество2 < > Количество, "Неправильное количество")
AS Вид_ошибки
FROM Список1 LEFT JOIN Список3 ON
Список1.ТД = Список3.ТД2
WHERE ТД2 Is Null OR Код_товара2 <
> Код_товара
OR Количество2 < > Количество
UNION SELECT Список1.ТД,
Список1.Код_товара, Список1.Количество,
Список3.ТД2, Список3.Код_товара2,
Список3.Количество2,
Switch(ТД2 Is Null OR ТД Is Null, "Неправильный номер ТД",
Код_товара2 < > Код_товара, "Неправильный код товара",
Количество2 < > Количество, "Неправильное количество")
AS Вид_ошибки
FROM Список1 RIGHT JOIN Список3
ON Список1.ТД = Список3.ТД2
WHERE ТД Is Null OR Код_товара2 <
> Код_товара
```

OR Количество2 < > Количество
 Второй вариант предполагает построение двух запросов на объединение:

- запроса, формирующего общий список из данных исходных таблиц;
- запроса на объединение результатов трех выборок.

К выборкам, которые нужно объединить, относятся:

- выборка по несоответствиям кода товара или количества,
- выборка из общего списка по критерию отсутствия декларации в таблице «Список2»,
- выборка из общего списка по критерию отсутствия декларации в таблице «Список1».

Первый запрос алгоритма предполагает объединение данных из таблиц «Список1» и «Список2» с одновременными преобразованиями в таблице «Список2» и построением вычисляемого поля «Список». Такое поле необходимо для обозначения, из какой таблицы был произведен импорт данных. Инструкция SQL для данного запроса представлена ниже:

```
SELECT Список1.ТД, Список1.Код_товара,
Список1.Количество, "Список1"
AS Список
FROM Список1
UNION SELECT КодТО & "/" &
Left(Дата,2) & Mid(Дата,4,2) &
Right(Дата,2) & "/" & Right("0000000" &
Номер_ТД,7)
AS ТД2, Right("0" & Код_товара,10)
AS Код_товара2, Список2.Количество2,
"Список2"
AS Список
FROM Список2
```

Второй запрос представляет собой запрос на объединение трех выборок, которые будут произведены непосредственно внутри него.

Первая выборка отражает данные, в которых имеются расхождения в полях «Код_товара» и «Количество». В запрос добавляются поля из таблиц «Список1» и «Список2», снова рассчитываются вычисляемые поля «ТД2» и «Код_товара2». Связь устанавливается между полем «ТД1», принятым как ключевое для таблицы «Список1», и вычисляемым полем «ТД2» таблицы «Список2». Данную связь можно описать только средствами языка SQL. Рассчитывается поле «Вид_ошибки» для двух возможных ошибок. В качестве условий отбора устанавливается неравенство между полями «Количество» или между полями «Код_товара» обеих таблиц.

Вторая выборка определяется на основе запроса «Общий_список» путем построения запроса с группировкой по полю «ТД». В полях «Код_товара», «Количество» и «Список» устанавливается функция First, в поле «ТД» – функция Count. Для попадания в выборку количество записей для поля «ТД» должно быть равно одной, т.е. в одном списке запись была, а во втором отсутствовала, и при этом это должна быть запись из таблицы «Список1». Для поля «Вид_ошибки» записывается постоянный комментарий о неправильном номере декларации. Пустые поля null добавлены для того, чтобы в итоговой выборке данные отражались

ТД	Код_товара	Количество	ТД2	Код_товара2	Количество2	Вид_ошибки
10009190/170120/0000022	8711100000	120	10009190/170120/0000022	8712003000	120	Неправильный код товара
10210000/171219/0000024	8516108000	300				Неправильный номер ТД
10216000/021219/0000002	0902300009	3500	10216000/021219/0000002	0902300009	2000	Неправильное количество

Рис. 7. Результат выполнения запроса на сравнение при отсутствии ключевых полей

именно в полях того списка, к которому они относятся.

Третья выборка формируется аналогично второй, но при этом отбираются только записи, принадлежащие таблице «Список2», меняется расположение пустых полей. Три выборки объединяются в запросе инструкциями UNION. Запись запроса на языке SQL представлена далее:

```
SELECT Список1.ТД, Список1.Код_товара,
Список1.Количество, КодТО & "/" &
Left(Дата,2) & Mid(Дата,4,2) & Right(Дата,2)
& "/" & Right("0000000" & Номер_ТД,7)
```

```
AS ТД2, Right("0" & Список2.Код_товара,10)
```

```
AS Код_товара2, Список2.Количество2,
Switch(Код_товара2<>Список1.Код_товара,
"Неправильный код товара",
Количество2<>Количество,
"Неправильное количество")
```

```
AS Вид_ошибки
```

```
FROM Список1
```

```
INNER JOIN Список2
```

```
ON Список1.ТД = Список2.КодТО & "/" &
Left(Дата,2) & Mid(Дата,4,2) & Right(Дата,2)
& "/" & Right("0000000" & Номер_ТД,7)
```

```
WHERE Right("0" & Список2.Код_товара,10)<>
Список1.Код_товара OR
Количество2<>Количество
```

```
UNION SELECT ТД, First(Код_товара)
```

```
AS ТД2, First(Количество)
```

```
AS Количество1, null, null, null, "Неправильный номер ТД"
```

```
AS Вид_ошибки
```

```
FROM Общий_список
```

```
GROUP BY ТД
```

```
HAVING (First(Список)="Список1") AND
(Count(ТД)=1)
```

```
UNION Select null, null, null, ТД
```

```
AS ТД2, First(Код_товара)
```

```
AS Код_товара2, First(Количество)
```

```
AS Количество2, "Неправильный номер ТД"
```

```
AS Вид_ошибки
```

```
FROM Общий_список
```

GROUP BY ТД

```
HAVING (First(Список)="Список2") AND
(Count(ТД)=1)
```

В результате выполнения запросов на объединение по обоим указанным способам отображается таблица, показанная на рис. 7.

Первые три поля таблицы имеют отношение к таблице «Список1», а следующие три поля относятся к таблице «Список2». Данная структура позволяет удобнее визуально сопоставлять данные. Пустые ячейки означают, что для данной декларации из одного списка не найдено аналогичной из другого списка, что означает необходимость дополнительной проверки обоих источников информации. Также в таблице отражены записи, в которых имеются расхождения по коду товара и по его количеству. По обоим предложенным способам решения результаты получаются одинаковые, что позволяет говорить о том, что способы взаимозаменяемы.

Таким образом, поставленная в работе задача решена. Составлены три алгоритма сравнения списков данных: с нежесткими и жесткими условиями, а также при наличии пропусков в записях. Во втором и третьем алгоритмах реализовано преобразование данных с помощью строковых функций и формирование комментариев с помощью функций управления.

Во втором алгоритме предполагается наличие ключевых полей в таблицах, отсутствие пропущенных записей. Вариант реализован запросом на выборку.

В третьем алгоритме предполагается отсутствие ключевых полей и наличие пропусков в записях. Вариант реализован двумя способами. Первый способ заключается в последовательном выполнении двух запросов: запроса на создание таблицы и запроса на объединение двух выборок. Второй способ реализован путем создания двух запросов: запроса на объединение

обоих списков в общий с указанием таблицы-источника и запроса на объединение трех выборок, позволяющих отобразить данные с несоответствиями.

Разработанные алгоритмы способны существенно ускорить и упростить работу таможенных инспекторов, в чьи обязанно-

сти входит сравнение больших списков данных, повысить эффективность их деятельности в процессе выявления фактов занижения таможенной стоимости товаров

Список литературы

1. Лавров М. Когда импортеру выгодно завышать или занижать цену ввозимого товара // Практическое налоговое планирование. № 9. Сентябрь. 2013. С.50–53.

2. Сальников В.И. Автоматизация решения задач для информационно-аналитического обеспечения деятельности таможенных органов с использованием SQL-запросов // Бюллетень инновационных технологий. 2019. Т.3. № 4(12).

3. Сальников В.И. Построение систем принятия решений на основе операций алгебры логики // Ученые записки Санкт-Петербургского имени В.Б. Бобкова филиала Российской таможенной академии. 2019. № 2 (70). С. 101–107.

Поступила в редакцию 20.01.2020

Сведения об авторе:

Сальников Владислав Игоревич – старший преподаватель кафедры таможенного дела Санкт-Петербургского филиала Российской таможенной академии, e-mail: visrta@yandex.ru

Электронный научно-практический журнал "**Бюллетень инновационных технологий**"
(ISSN 2520-2839) является сетевым средством массовой информации
регистрационный номер Эл № ФС77-73203
по вопросам публикации в Журнале обращайтесь по адресу bitjournal@yandex.ru