Титульник

Реферат

**Содержание**

[Введение 5](#_Toc84061580)

[1 Анализ существующих аналогов 6](#_Toc84061581)

[2 Техническое задание 6](#_Toc84061582)

[3 Разработка парсера Excel файла 6](#_Toc84061583)

[4 Разработка исполнителя SQL-запросов 11](#_Toc84061584)

[5 Разработка пользовательского интерфейса 11](#_Toc84061585)

# Введение

# Анализ существующих аналогов

# Техническое задание

# Разработка парсера Excel-файла

Парсинг – это метод индексирования информации с последующей конвертацией ее в иной формат или даже иной тип данных. Парсинг позволяет взять файл в одном формате и преобразовать его данные в более удобоваримую форму, которую можно использовать в своих целях.

В нашем случае парсинг это процесс обработки данных из Excel-файла. Он подразумевает анализ текста, вычленение оттуда необходимых материалов и их преобразование в подходящий вид. Благодаря парсингу можно находить на страницах файла небольшие клочки полезной информации и в автоматическом режиме их оттуда извлекать, чтобы потом переиспользовать.

Вычленяемая информация в нашем случае это будет та информация, которая нужна для вставки в SQL-скрипт. Представлена она может быть как отдельными полями, так и табличными значениями.

## 3.1 Дескриптор Excel-файла общая информация

Под дескриптором Excel-файла будем понимать ту информацию, которая описывает извлекаемые объекты из файла. Дескриптор состоит из объектов и их атрибутов. На данный момент есть три вида объектов, которые заключаются в открывающий и закрывающий тег:

* <singleValue>…</singleValue> - информация об значении, хранившемся в одном поле;
* <table>…</table> - информация об таблице, в которой хранятся значения;
* <column>…</column> - является вложенным тегом в тег table и является информацией о столбце таблицы, в котором хранятся данные.

Атрибуты в свою очередь представляются в виде токенов. Под токеном подразумевается значение и короткого описания этого значения (имя токена). Токен представляется следующим образом:

Имя токена : значение токена;

Каждый токен объекта должен быть отделен точкой с запятой, при этом, если имя или значение имеют пробелы и табуляцию, то они должны быть заключены в кавычки. Таким образом дескриптор объекта, который имеет N атрибутов и один вложенный тег, выглядит следующим образом:

<tag>

nameToken\_1 : valueToken\_1;

nameToken\_2 : valueToken\_2;

…

nameToken\_N : valueToken\_N;

<nestedObject>

nameToken\_1 : valueToken\_1;

nameToken\_2 : valueToken\_2;

…

nameToken\_N : valueToken\_N;

</nestedObject>

</tag>

В таблице 1 представлены возможные имена атрибутов тега singleValue.

Таблица 1 – Возможные атрибуты тега singleValue

|  |  |  |  |
| --- | --- | --- | --- |
| Коды атрибутов | Допустимые значения | Описание | Если значение не указано |
| SHEET\_NUMBER | Целочисленные числа, больше 0 | Номер страницы, на которой производится поиск (отсчет страниц ведется с 1). | Поиск производится на всех страницах |
| SHEET\_NAME | Любая строка | Имя страницы, на которой производится поиск. | Поиск производится на всех страницах |
| SECTION\_NAME | Любая строка | Раздел на странице, относительно которого производится поиск (для полей с одинаковыми надписями). | Поиск производится на всей странице |
| SECTION\_BOTTOM\_LEFT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область снизу слева от раздела включать в поиск, рисунок 1. | Поиск не производится в левой нижней части от раздела (Аналогично значению 0). |
| SECTION\_BOTTOM\_RIGHT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область снизу справа от раздела включать в поиск, рисунок 1. | Поиск не производится в правой нижней части от раздела (Аналогично значению 0). |
| SECTION\_UP\_LEFT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область сверху слева от раздела включать в поиск, рисунок 1. | Поиск не производится в левой верхней части от раздела (Аналогично значению 0). |
| SECTION\_UP\_RIGHT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область сверху справа от раздела включать в поиск, рисунок 1. | Поиск не производится в правой верхней части от раздела (Аналогично значению 0). |
| FIELD | Любая строка | Текст ячейки, относительно которой ищется значение. | Аналогично пустой ячейке. |
| CODE | Любая строка | Код значения, по которому можно идентифицировать данное поле. | Значение данного объекта будет получено, но обратиться к нему будет нельзя в SQL-скрипте. |
| OFFEST\_ROW | Любое целочисленное число | Смещение по строке относительно FIELD. | Аналогично значению 0 |
| OFFEST\_COLUMN | Любое целочисленное число | Смещение по столбцу относительно FIELD. | Аналогично значению 0 |

В таблице 2 представлены возможные имена атрибутов тега table.

Таблица 2– Возможные атрибуты тега table

|  |  |  |  |
| --- | --- | --- | --- |
| SHEET\_NUMBER | Целочисленные числа, больше 0 | Номер страницы, на которой производится поиск (отсчет страниц ведется с 1). | Поиск производится на всех страницах |
| SHEET\_NAME | Любая строка | Имя страницы, на которой производится поиск. | Поиск производится на всех страницах |
| SECTION\_NAME | Любая строка | Раздел на странице, относительно которого производится поиск (для полей с одинаковыми надписями). | Поиск производится на всей странице |
| SECTION\_BOTTOM\_LEFT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область снизу слева от раздела включать в поиск, рисунок 1. | Поиск не производится в левой нижней части от раздела (Аналогично значению 0). |
| SECTION\_BOTTOM\_RIGHT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область снизу справа от раздела включать в поиск, рисунок 1. | Поиск не производится в правой нижней части от раздела (Аналогично значению 0). |
| SECTION\_UP\_LEFT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область сверху слева от раздела включать в поиск, рисунок 1. | Поиск не производится в левой верхней части от раздела (Аналогично значению 0). |
| SECTION\_UP\_RIGHT | 1 или 0 | При условии, что атрибут SECTION\_NAME определен. Область сверху справа от раздела включать в поиск, рисунок 1. | Поиск не производится в правой верхней части от раздела (Аналогично значению 0). |
| CODE | Любая строка | Код таблицы, по которому можно идентифицировать данную таблицу. | Значения данной таблицы будут получены, но обратиться к ним будет нельзя в SQL-скрипте. |
| INCLUDE\_FINAL\_ROW | 1 или 0 | Записывать последнюю строку таблицы, которая определяется во вложенных объектах column. | Аналогично значению 0 |

В таблице 3 представлены возможные имена атрибутов тега column.

Таблица 3– Возможные атрибуты тега table

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | Любая строка | Заголовок столбца. | Поиск производится на всех страницах |
| CODE | Любая строка | Код столбца, по которому можно идентифицировать данный столбец. | Значения данного столбца будут получены, но обратиться к ним будет нельзя в SQL-скрипте. |
| FINAL\_CELL | Любая строка | Значение, говорящее, что данная ячейка последняя. Последней строкой в таблице считается та строка, когда у всех столбцов соответствующая ячейка равняется ее конечному значению. | Любой текст в ячейке является сигналом конечной строки. |

На рисунке 1 изображены области для поиска относительно раздела при соответственно установленных флагов.



Рисунок 1 – Области для поиска относительно раздела

## 3.2 Парсинг дескриптора Excel-файла

Парсинг дескриптора Excel-файла происходит следующим образом. Происходит проход по дескриптору, пока не будет найден значащий символ (любой символ кроме табуляции и пробелов). Далее определяется, является ли данный символ началом тега. Если нет, то выдается ошибка, иначе читается имя тега и происходит дальнейший проход. После тега так же ищется первый значащий символ. Далее определяется, является ли данный символ началом вложенного тега или окончанием текущего тега. Если ни тем, ни другим не является, то считается, что встретили атрибут объекта. Для атриубта берется подстрока от текущей позиции до символа точки с запятой, и из данной подстроки извлекается пара имени и значения токена, после чего прочитанный токен помещается в список токенов текущего объекта. Если же встретился вложенный тег, то обработка его происходит аналогично тому, как было описано выше, только после встречи закрывающего тега вложенного объекта происходит добавление прочитанного вложенного объекта в список вложенных объектов того объекта, в котором этот вложенный объект был встречен. Если же встречен закрывающий тег текущего объекта, то данный объект помещается в список дескриптора объекта и происходит дальнейший поиск тега, либо окончания дескриптора Excel-файла. Описанный алгоритм отображен на рисунке 2.



Рисунок 2 – Алгоритм получения дескриптора объекта

# Разработка исполнителя SQL-запросов

Задача исполнителя SQL-запросов заключается в том, чтобы занести прочитанные данные из Excel-файла в готовый SQL-запрос и отправить на сервер отредактированный запрос на выполнение.

Место для вставки одиночных данных в SQL-запрос должно представлять из себя код одиночного значения, заключенного в символы решетки. Например, была получена дата из Excel файла, которой был присвоен код dateReport. Место, в которое должно быть вставлено это значение, должно выглядеть следующим образом:

DECLARE @date = CONVERT(DATE, '#dateReport#')

И тогда, перед тем, как исполнить SQL-запрос, данный код будет заменен прочитанным значением, например 01-01-2021:

DECLARE @date DATE = CONVERT(DATE, '01-01-2021')

Для табличных значений необходимо на место, где они должны быть вставлены, вставить шаблон строки, которая будет копироваться и вставляться на место данного шаблона. При этом шаблон должен быть заключен в символы «<» и «>» и в начале иметь код таблицы, а далее в скобках указаны столбцы таблицы, заключенные в символы решетки. Допустим была прочитана таблица, код для которой был указан dataForInsert. Данная таблица имеет два столбца с кодами dateInfo и Info, тогда шаблон для вставки данных значений в SQL-запрос будет выглядеть следующим образом:

<dataForInsert (CONVERT(DATE, '#dateInfo#'), '#Info#')>

И тогда, перед тем, как исполнить SQL-запрос, данный шаблон будет заменен прочитанными значениями, например:

(CONVERT(DATE, '01-01-2021'), 'Новый год')

(CONVERT(DATE, '23-02-2021'), 'День защитника отечества')

При составлении SQL-запроса нужно учитывать, что при вставки значений не учитываются типы данных, поэтому, если данные должны быть вставлены как строка, то необходимо код, заключенный в символы решетки, заключить еще в одинарные кавычки. А если, к примеру, данные должны быть вставлены как дата, то нужно еще поместить шаблон в конструкцию по конвертации строки в дату, как это было показано выше.

# Разработка пользовательского интерфейса