Программа «Предложения поставщиков» (TTLoader)

Преднозначена для загрузки предложений поставщиков в выбранную тендорную таблицу(ТТ).

Основная функция – избавиться от рутинных операций сбора данных перед загрузкой. Во-первых, это крайне утомительный процесс с кучей мелочей, которые легко упустить из виду. Во-вторых, старый подход не кошерен и не универсален в принципе.

**Функционал программы**

- загрузка данных «на лету» из исходного файла поставщика. Без сбора их в объемный общий файл с перестройкой структуры данных. Так же, пакетная загрузка файлов – просто указать папку;

- автоматическое расширение структуры данных при парсинге документа. Например, добавление новых маршрутов в группу (что крайне мало вероятно, поскольку данные предоставляются в шаблонах приложенных к договору), добавление новой группы цен;

- загрузка данных в режимах: дополнения и перезаписи;

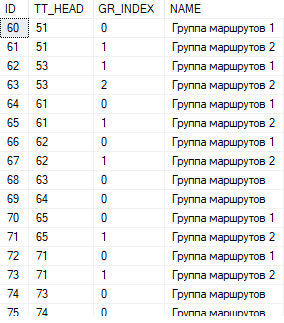
- удаление данных выбранного поставщика;

- гибкий парсинг исходной формы, чтобы не переписывать программу каждый год.

**Описание структуры данных**

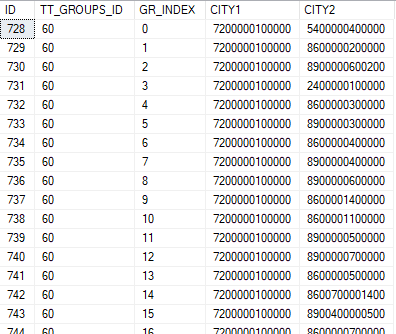
ТТ создается в программе «Тендерные таблицы»(TenderTable). Заглавная таблица **nftTU.TT\_HEAD**.

Данная программа не может создавать тендеры, но привязывает к ней данные.

**TT\_GROUPS**

Далее, при загрузке создаются группы маршрутов. На данный момент их две: из Тюмени по области, и из ключевых городов страны в Тюмень (См. приложение 1). Данные группы хранятся в таблице **nftTU.TT\_GROUPS** и привязаны к записям **nftTU.TT\_HEAD**.

На данный момент, группа 1 = из тюмени по области, группа 2 = в Тюмень из любой точки страны.

**TT\_MARSHRUT**

К группам маршрутов привязан набор конкретных маршрутов. **Где TT\_GROUPS\_ID = TT\_GROUPS.ID**

Данные маршруты в текстовом виде присутствуют в excel файле с ценами поставщика. В базе нет данных содержfщих эти маршруты в текстовом виде «Город1-Город2», но индексы **CITY1** и **CITY2** содержатся в справочнике **CLADR\_KLADR.CODE,** что может позволить идентифицировать маршрут распарсив строку и найдя нужные города в справочнике. Но есть сложность – города разделяются дефисом и есть наименования городов с дефисом.

Поле **GR\_INDEX,** скорее всего, используется при сортировке по алфавиту. В исходных файлах с ценами не содержится. Можно ориентироваться по порядку следования в строках.

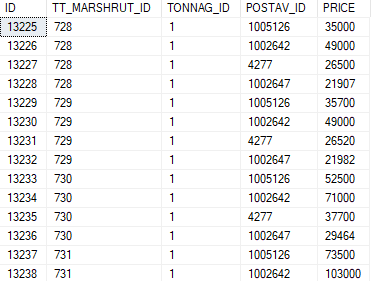
**TT\_DATA**

Таблица привязана к конкретному маршруту и содержит цену на перевозку с конкретным тоннажом.

Тоннаж представляет собой значение справочника **nft\_TU.TONNAG**, где к id привязаны значение типа «От 5,0 до 10,0 тонн» или «Негабаритный с высотой 5 м. от 10,0 до 18,0 тонн». Данные текстовые значения встречаются в заголовках столбцов документа.

Поставщик – общее для всего документа значение, которое выбирается пользователем из справочника **nft.Postav** перед загрузкой данных.

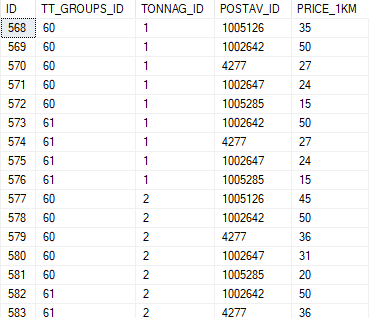
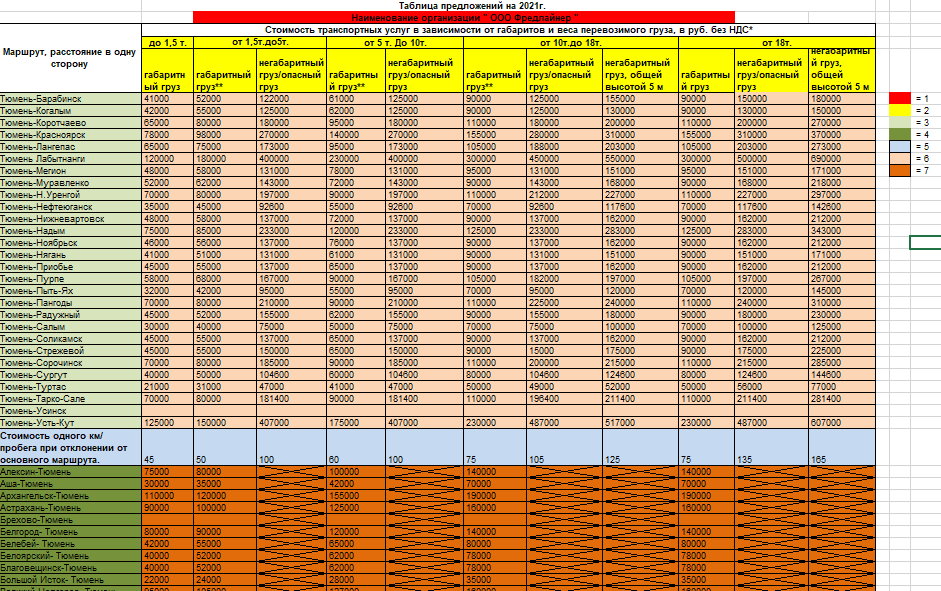
**TT\_OTKLON**

Таблица содержит стоимость за километр отклонения от рассчитанного маршрута. Стоимость цельного обычного маршрута оценена в таблице TT\_DATA, но за сверурочный километраж наценка лежит здесь.

Если поставщик не указал цену отклонения, она тут не хранится.

**Сопоставление БД и входного файла**

****

1 – Поле с наименованием поставщика. Может быть пустым. Не годится для надежной идентификации, требуется ручной выбор пользователя. Определяет **POSTAV\_ID** для всего документа.

2 – набор текстовых маркеров для выбора **ID** типа тоннажа из справочника **TONNAG** для каждой колонки (6) и (7). Хоть форма и стандартная, нельзя исключать «рукоблудства» поставщика, который может поменять текстовку. После проведения текстового анализа документа, если не удастся идентифицировать некоторые поля, даем пользователю возможность вручную указывать значения.

3 – первая группа маршрутов. Идентифицируется по первому городу «Тюмень». Каждое значение должно быть сопоставлено с записью из таблицы **TT\_MARSHRUT**.

4 – вторая группа маршрутов. Идентифицируется вторым городом «Тюмень». Каждое значение должно быть сопоставлено с записью из таблицы **TT\_MARSHRUT**.

5 – сторка отклонений для предыдущей группы маршрутов. Непустые значения заносятся в таблицу **TT\_OTKLON**. Строка есть после каждой группы маршрутов, включая (4).

6,7 – данные для **TT\_DATA**, в привязке к маршруту (3) или (4) этой строчки.

**Реализация открытия файла**

При открытии файла excel он используется не в чистом виде, а загоняется в двумерный массив для дальнейшего анализа и модификации. Так же в этом массиве добавляются элементы для хранения id распознанных элементов. Таким образом ячейка таблицы является вариативной записью для представления следующих типов:

- *маршрут*. Содержит: ***text***, ***group\_id***, ***marshrut\_id***. Распространяет значение ***marshrut\_id*** на всю строку. Распространяет ***group\_id*** на всю колонку до ближайшей ячейки *отклонения* и включает ее.

- *тоннаж*. Содержит: ***tonnag\_id***. Распространяет значение на всю колонку.

- *отклонение*. Содержит ***price***, ***group\_id***, ***tonnag\_id***. Распространяет ***group\_id*** и свой *тип* на всю строку.

- *данные*. Содержит: ***price, marshrut\_id***, ***tonnag\_id***.

После открытия файла формируется двухмерный массив записей.

*Этап 1. Поиск маршрутов.*

Ячейки файла анализируется слева направа, сверху вниз.

Исходя из логики документа, сначала мы должны обнаружить колонку маршрутов и отклонений.

Проанализировав ее и раздав ячейкам типы со значениями text, group\_id, marshrut\_id, получаем высоту таблицы.

*Этап 2. Поиск тоннажей.*

Начинаем анализ строк выше начала колонки маршрутов – там находятся признаки тоннажа.

Обнаружив их, распознав и раздав значения tonnage\_id, получаем ширину таблицы.

*Этап 3. Данные.*

Все, что находится правее маршрутов и ниже тоннажа – данные.

Забиваем данные в ячейки массива и указываем тип ячеек.

*Этап 4. Экстрополяция отклонений и маршрутов.*

Перебираем колонку маршрутов и в строках, где попадаются ячейки отклонения – меняем тип всех ячеек на всю ширину таблицы.

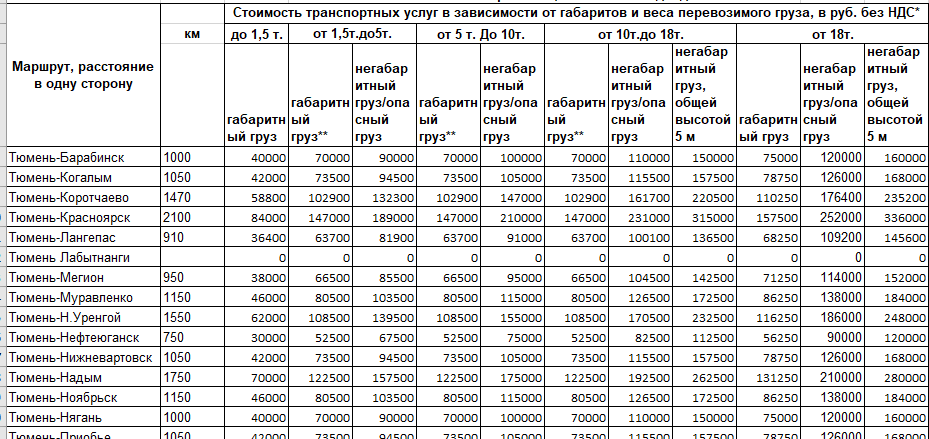
Для ячеек маршрутов – заносим marshrut\_id во все ячейки данных на всю ширину таблицы.

Этап 6. *Экстрополяция тоннажа.*

Перебираем все ячейки тоннажа и заносим значение tonnag\_id во все ячейки столбца на всю высоту таблицы.

В итоге мы получаем полную матрицу, где мы в каждой ячейке имеем тип и все необходиме для этого типа данные для внесения в базу. Т.е. можно не только полностью проконтролировать данные ячеек отдельно друг от друга, но и вносить данные в базу в произвольном порядке и визуализировать каждую ячейку индивидуально, без отслеживания ее привязки по положению и т.п.

Кроме того, это позволит отсечь «некачественные» ячейки. Например:



Здесь поставщик решил добавить колонку с километражом, а стоимость вычислить формулами.

Колонка километров будет не качественной колонкой данных, поскольку там будут отсутствовать tonnage\_id. При импорте она будет проигнорирована и данные будут корректны.

Приложение 1. Маршруты на 2021 год

|  |  |
| --- | --- |
| Группа 1 (GR\_INDEX = 0)  Тюмень-Барабинск  Тюмень-Когалым  Тюмень-Коротчаево  Тюмень-Красноярск  Тюмень-Лангепас  Тюмень Лабытнанги  Тюмень-Мегион  Тюмень-Муравленко  Тюмень-Н.Уренгой  Тюмень-Нефтеюганск  Тюмень-Нижневартовск  Тюмень-Надым  Тюмень-Ноябрьск  Тюмень-Нягань  Тюмень-Приобье  Тюмень-Пурпе  Тюмень-Пыть-Ях  Тюмень-Пангоды  Тюмень-Радужный  Тюмень-Салым  Тюмень-Соликамск  Тюмень-Стрежевой  Тюмень-Сорочинск  Тюмень-Сургут  Тюмень-Туртас  Тюмень-Тарко-Сале  Тюмень-Усинск  Тюмень-Усть-Кут  Группа 2 (GR\_INDEX = 1)  Алексин-Тюмень  Аша-Тюмень  Архангельск-Тюмень  Астрахань-Тюмень  Брехово-Тюмень  Белгород- Тюмень  Белебей- Тюмень  Белоярский- Тюмень  Благовещинск-Тюмень  Большой Исток- Тюмень  Великий Новгород- Тюмень  Воронеж-Тюмень  Волгоград-Тюмень  Воткинск-Тюмень  Владимир-Тюмень  Гагарин-Тюмень  Гороховец- Тюмень  Гусь-Хрустальный-Тюмень | Димитровград-Тюмень Екатеринбург-Тюмень  Златоуст- Тюмень  Заречный-Тюмень  Ижевск-Тюмень  Казань- Тюмень  Каменск-Уральск-Тюмень  Катайск- Тюмень  Котельники-Тюмень  Курган-Тюмень  Кыштым-Тюмень  Лиски-Тюмень  Ливны-Тюмень  Лобня- Тюмень  Лысьва-Тюмень  Магнитогорск-Тюмень  Миасс- Тюмень  Москва-Тюмень  Муром-Тюмень  Набр- Челны- Тюмень  Новосибирск-Тюмень  с.Нагорное-Тюмень  Озерск- Тюмень  Октябрьский- Тюмень  Омск-Тюмень  Орел-Тюмень  Оренбург-Тюмень  Отрадный-Тюмень  Орск-Тюмень  п.Красногвардейский-Тюмень  Пенза-Тюмень  Первоуральск- Тюмень  Пермь-Тюмень  Полевской- Тюмень  р.п. Пышма-Тюмень  Рязань-Тюмень  Ростов-на-Дону-Тюмень  Самара- Тюмень  Санкт-Петербург-Тюмень  Саратов-Тюмень  Сафоново-Тюмень  Томск-Тюмень  Тула-Тюмень  Уфа-Тюмень  Чебоксары- Тюмень  Челябинск-Тюмень  Чусовой- Тюмень  Чернушка-Тюмень  Шадринск- Тюмень  Ярославль-Тюмень  Ясногорск- Тюмень |