

Вводные:

1. Есть таблица Адреса - Населенный пункт | Улица | Дом
2. Нужно сопоставить с ней другую таблицу адресов

Алгоритм:

1. Загрузить таблицу адресов, например в формате Excel 97-2003
2. Определить сопоставление колонок данных (желательно автоматически). Наличие полей произвольно и зависит от провайдера. Минимальные необходимые - улица и дом, например провайдер только в одном городе:
 - 2.1. Провайдер (проставить по всем вставляемым записями)
 - 2.2. НП+Улица+Дом (сводный адрес)
 - 2.3. Улица+Дом (сводный адрес без НП, НП нужно спросить)
 - 2.4. Основные поля отдельно
 - 2.4.1. НП (с уникализатором области/района/микрорайона)
 - 2.4.2. Улица (с типом)
 - 2.4.3. Дом (с корпусом)
 - 2.5. Дополнительные поля
 - 2.5.1. Область
 - 2.5.2. Район области
 - 2.5.3. Тип НП (город, село, поселок, ПГТ)
 - 2.5.4. Район НП
 - 2.5.5. Микрорайон
 - 2.5.6. Улица без типа
 - 2.5.7. Тип Улицы
 - 2.5.8. Другое название улицы (старое, новое, альтернативное)
 - 2.5.9. Дом без корпуса
 - 2.5.10. Корпус
3. Запросить недостающие данные (например нет НП), проставить его по всей таблице
4. Применить легкие скрипты очистки и нормализации данных
 - 4.1. Двойные и начальные/завершающие пробелы
 - 4.2. Двойные точки, двойные запятые
5. Сопоставить с базой адресов (нормализованный сводный адрес сравнить с аналогами из справочника,
6. Для оставшихся, применить скрипты очистки и нормализации данных:
 - 6.1. Латинские символы
 - 6.2. Нормализация номеров домов (есть макрос для Excel)
 - 6.3. Нормализация типов улиц (есть макрос для Excel)
7. Сопоставить с базой адресов
8. Для оставшихся, провести сопоставление с таблицей сопоставления по предыдущим загрузкам покрытия
9. Сопоставить с базой адресов
10. Для оставшихся, применить геокодирование:
 - 10.1. Google - <http://maps.google.com/maps/api/geocode/xml?address={строка}&sensor=true&language=ru>
 - 10.2. Yandex- <http://geocode-maps.yandex.ru/1.x/?geocode={строка}>

10.3. Visicom, для адреса -
http://api.visicom.ua/data-api/1.0/ru/search/adr_address.json?text={строка}&key={API ключ, предоставляю}, для улицы -
http://api.visicom.ua/data-api/1.0/ru/search/adr_street.json?text={строка}&key={API ключ, предоставляю}

10.4. ...

11. Если геокодирование не возвращает результата в том же населенном пункте, то кодируем по частям - сначала НП, потом НП+Ул, потом НП+Ул+Дом
12. Сопоставить с базой адресов
13. Для оставшихся, применить неточное сопоставление с минимальным % несоответствия
14. Сопоставить с базой адресов
15. Для оставшихся, вывести список для ручного резолвинга, применить неточное сопоставление со средним или высоким % несоответствия.
16. Необходима возможность ручного добавления адресов в справочник на этапе ручного сопоставления
17. По достижению 100% сопоставления загруженного покрытия, сравнить с текущим покрытием провайдера:
 - 17.1. Новые адреса (которых ранее не было) - добавить в покрытие
 - 17.2. Удаленные адреса (которые были но теперь нет) - вывести для подтверждения удаления

Уникализаторы - используются для уникализации НП и Улиц. Например:

1. Алексеевка (Макаровский р-н)
2. Садовая ул. (Дарница)

Для уникализации используются родительские сущности, которые различаются для 2-х записей одного уровня, но не уникальных, в рамках:

1. Населенные пункты - в рамках Украины
2. Улицы - в рамках населенного пункта

Уникализаторы добавляются в самом конце, в скобках, по такой логике:

1. Область - например (Киевская обл.). Андреевка, Киевская область = Андреевка (Киевская обл.)
2. Район - например (Макаровский р-н), (Шевченковский р-н). Новоселки, Киево-Святошинский район = Новоселки (Киево-Святошинский р-н)
3. Микрорайон - например (Дарница). Боженко ул., Троещина микрорайон = Боженко ул. (Троещина)
4. Сельсовет, горсовет - например (Андровский с-т), (Одесский г-т). Полоузовка, Андровский сельсовет = Полоузовка, (Андровский с-т)

Добавление уникализатора для НП:

условие	Действие
Проверить уникальность названий	уникальные оставляем как есть, для неуникальных следующее условие

Проверить, есть ли среди неуникальных названий один НП более высокого уровня - город или ПГТ (например город Ялта, Львов в отличии от сел)	оставляем их как есть, для остальных следующее условие
Проверить, какие из НП станут уникальные, если добавить название области в скобках, включая НП более высокого уровня (например "Алексеевка (Крым)". Не подходит, если 2 неуникальных НП разного уровня в одной области.	оставляем с областью, для остальных следующее условие
Проверить, какие из НП станут уникальные, если добавить название района в скобках, включая НП более высокого уровня (например "Алексеевка (Броварской)". Не подходит, если 2 неуникальных НП разного уровня в одном районе.	оставляем с районом, для остальных следующее условие
Проверить, какие из НП станут уникальные, если добавить название сельсовета в скобках, включая НП более высокого уровня (например "Алексеевка (Алексеевский)". Не подходит, если 2 неуникальных НП разного уровня в одном районе.	оставляем с сельсоветом, для остальных следующее условие
То же самое для Область + Район (Киевская, Броварской) Район+Сельсовет (Броварской, Дибровский) Область+Район+Сельсовет (Киевская, Броварской, Дибровский)	
Проверить, какие из НП станут уникальные, если добавить тип НП в скобках (например "Краснодон (пгт)". Подходит для 2-х неуникальных НП разного уровня в одном районе.	

После реализации древовидной таблицы:

1. Хранение справочника адресов в таком виде

2. Таблица сопоставления старых результатов в формате НП-НП, НП+Ул-НП+Ул, а не только полный адрес.

Расшифровка статусов ID:

-1 нет ответа геокодера (не совпали НП, тип улицы и номер дома)

0 не совпал после нормализации

1 новый адрес в базу для загрузки

Алгоритм действий после получения результатов обработки несопоставленных адресов геокодером Google:

Исходный адрес совпал с ответом, точность - ROOFTOP	Автоматически добавляем адрес в новые
Исходный адрес совпал с ответом по НП, типу улицы и номеру дома. При этом в исходном адресе и ответе нет номера (например 2-й бортницкий). После замены адрес совпал со справочником.	Заменяем название улицы в текущем адресе и других адресах по этой улице, проверяем сопоставление.