Daten sichten, bereinigen und integrieren mit OpenRefine

Ein Arbeitsblatt für Studenten von Data Science/Data Engineering und Interessierte

1. Einleitung

Ziele

Dieses Arbeitsblatt zeigt, wie Sie OpenRefine, ein Desktop-Tool für die Datenvorverarbeitung ohne Programmierung (Low-Coding), verwenden können. Es werden einige Möglichkeiten diskutiert mit Daten umzugehen.

Bei OpenRefine handelt es sich um ein Tool zum Sichten ('Explorieren'), Bereinigen (Aufarbeiten/Säubern), Integrieren und Anreichern von Daten.

Die Aufgaben bestehen aus realistischen Beispielen, welche das Arbeiten mit heterogenen Daten beinhaltet. Dabei soll aufgezeigt werden, wie man mit diesen Daten umzugehen hat, wenn man mit Problemen aus diesem Bereich konfrontiert wird.

Die Ziele dieses Arbeitsblattes sind:

- **OpenRefine** verstehen.
- OpenRefine nutzen, um **Daten zu sichten**.
- Herausforderungen verstehen, die heterogene Daten mit sich bringen können und OpenRefine nutzen, um Daten zu **bereinigen**.
- OpenRefine nutzen, um zwei Datensätze zu **zusammenzuführen**.
- OpenRefine zur **Anreicherung** von Daten mithilfe eines Geokodierungsdienstes verwenden.
- OpenRefine verwenden, um auf **strukturierte Daten** zuzugreifen, indem eine Webseite gescraped wird.

Zeitplanung

Ungefähr eine Stunde für den Leseteil (ohne Aufgaben), zusätzlich ca. 1 Stunde und 15 Minuten für die Aufgaben - beide Angaben können je nach Wissensstand abweichen.

Voraussetzungen

Um das Arbeitsblatt durcharbeiten zu können, brauchen Sie folgende Dinge:

- Hardware: Mindestens 1 GB an unbenutzten RAM
- Internetzugang (zum Herunterladen von der benötigten Software und Daten - zusätzlich für die fortgeschrittenen Kapitel)
- Software: OpenRefine (verfügbar für Windows, Mac und Linux) zu installieren wie unten beschrieben.
- Daten: OpenRefine Aufgabenblatt Daten.

Folgende Themen können als Vorbereitung auf dieses Thema hilfreich sein:

- Grundlegendes Verständnis im Umgang mit Daten
- Grundlegendes Wissen über Datentypen
- Grundlagen in SQL

Installation von OpenRefine

Um die Aufgaben durchführen zu können, muss OpenRefine auf der lokalen Maschine installiert werden (OpenRefine kann auch im Rahmen eines Kurses zur Verfügung gestellt werden). Für Windows wird die "Windows kit with embedded Java" Version empfohlen, da man mit dieser auf eine Java-Installation verzichten kann. Auf Linux wird unabhängig von der Version eine Javainstallation vorausgesetzt. Getestete Browser für OpenRefine sind Firefox, Chrome und Safari. Internet Explorer und Edge sind nicht unterstützt. Eine Anleitung zur Installation von OpenRefine ist auf der OpenRefine Website zu finden.

Wenn Sie auf Probleme stossen sollten, ist das FAQ in der Anleitung eine erste Anlaufstelle. Alternativ können Sie auch direkt in der Bedienungsanleitung nach Lösungen suchen. Siehe auch diese Help-Page.

Struktur des Arbeitsblatts

- Im nächsten Kapitel 2 werden die Grundlagen der Datenintegration und Datenanreicherung erläutert.
- In den Kapiteln 3 und 4 werden die Grundlagen und die Funktionsweise von OpenRefine erklärt.
- In Kapitel 5 wird erklärt, was Geokodierung ist und man diese in OpenRefine nutzen kann.
- In Kapitel 6 wird eine Wikipedia-Webseite mit OpenRefine gescraped.
- Die Übungen in den Kapiteln 4, 5 und 6 dienen dazu, die Kenntnisse über die Nutzung von OpenRefine zu vertiefen, indem ein ganzer Arbeitsablauf mit OpenRefine durchlaufen wird.

2. Über Datenintegration und Datenanreicherung

Daten liegen im Wesentlichen entweder in strukturierter oder unstrukturierter Form vor. Mit **strukturierten** Daten meint man modellierte Daten mit einem zugewiesenen Datentyp. Eine Adressliste (in einer CSV-Datei) wird typischerweise als strukturierte Daten klassifiziert. **Unstrukturierte Daten** sind hingegen Daten welche z.B. als Fliesstext gespeichert werden, wie ein Roman. Mit dieser Art von Adressliste ist es unmöglich z.B. einen Serienbrief zu versenden. Aber selbst bei strukturierten Daten unterscheidet man zwischen heterogenen und potentiell inkonsistenten Daten. Mit **inkonsistenten Daten** meint man Daten mit vielen fehlenden Einträgen oder Daten welche im "falschen" Format oder in der "falschen" Form gespeichert wurden - zumindest im Rahmen dieses Arbeitsblatts.

Datenintegration ist der Prozess des Vereinens von Daten aus verschiedenen Quellen zu einer einheitlichen Form. Der Integrationsprozess beginnt mit dem Importieren und beinhaltet Schritte wie Säubern, "Schema-Mapping" und Transformieren. OpenRefine bietet dabei viele Werkzeuge an, welche hilfreich sein können für das Integrieren von Daten (mehr dazu in den Kapitel 3 und 4).

Ein typischer Integrationsprozess ist, wenn zwei Adresslisten unterschiedlicher, unabhängiger Quellen zusammengeführt werden sollen. Diesen Prozess werden wir in einer Aufgabe üben (mehr dazu im Kapitel 4).

Datenanreicherung ist ein Begriff, der für das "Verbessern" und Komplettieren bereits existierender Daten steht. Dies kann auf verschiedenen Wegen passieren, wie z.B. Verbinden mit neuen Daten oder durch das Einbinden von externen Daten. Ein Beispiel dafür wäre das Hinzufügen von Koordinaten aufgrund von Postadressen. Dieser Prozess wird **Geocoding** genannt und erlaubt das Darstellen von Adressen auf einer Karte (mehr dazu im Kapitel 5).

Suchmaschinen setzen Crawler (oder Bots) ein, um das Web zu "scrapen". **Web-Scraping** ist der Prozess der Extraktion von Daten/Inhalten aus einer Website zur Analyse oder anderen Verwendung. Dabei wird das HTML-File einer Website eingelesen und nach Daten gefiltert. Ein Beispiel dazu kommt später als Aufgabe vor (siehe Kapitel 6).

3. OpenRefine Basics

OpenRefine ist ein Anwendungstool, das lokal als eigenständige Anwendung auf Ihrem Computer läuft und einen Webbrowser als Benutzeroberfläche verwendet. OpenRefine ist in Java geschrieben und als Open-Source Projekt auf GitHub verfügbar und stammt aus der Verwaltung von Wissensdatenbanken, wie Wikidata. Es gibt übrigens auch Open-Source-Alternativen, die ähnliche Konzepte verfolgen, wie Workbench (in Python geschrieben).

OpenRefine liest und verwaltet grosse Mengen an tabellarischen Daten, die typischerweise unübersichtlich und unstrukturiert sind. Für die meisten seiner Funktionen benötigt OpenRefine keinen Internetzugang, da es lokal ausgeführt wird. Das führt dazu, dass die Daten ebenfalls alle lokal abgespeichert werden.

Unter Bezugnahme auf die in Kapitel 2 erwähnten generischen Aufgaben kann OpenRefine

folgendes tun:

- Untersuchung der Daten, z.B. Facettierung und Clustering.
- Bereinigung unübersichtlicher Daten, z.B. unstrukturierter (oder halb-strukturierter) Textdateien.
- Datentransformation - Massentransformation von Daten, z.B. Datennormalisierung, Datenformatierung.
- Datenvalidierung und Deduplizierung.
- Datenabgleich mit externen Diensten, wie z.B. Wikidata.
- Zugriff auf Website-Daten.

OpenRefine ist eine Desktopapplikation, die als Client-Server implementiert ist, mit einem Browser als Client. Das Prinzip von OpenRefine ist, dass Daten von einer Quelle (Datei, Service, Datenbank) gelesen werden und als Kopie - d.h. als eigenes OpenRefine Projekt - lokal gespeichert werden.

Die Benutzeroberfläche (GUI)

OpenRefine bietet eine interaktive grafische Benutzeroberfläche (englisch: Graphical User Interface, GUI), die jeden Schritt der Arbeit mit Ihrem Datensatz visualisiert. Sie besteht aus einem Hauptfenster, in welchem die Daten angezeigt werden, an denen Sie gerade arbeiten.

Diese Tabelle ändert sich jedes Mal, wenn Sie eine Aktion wie Filtern oder Erstellen verschiedener Facetten durchführen. Der linke Teil der Benutzeroberfläche enthält eine Registerkarte, auf der sämtliche Facetten/Filter angezeigt werden, die sich derzeit auf den Datensatz auswirken. Hier können bereits erstellte Facetten und Filter angepasst werden. Jede Änderung wird direkt auf dem Datenblatt angewendet.

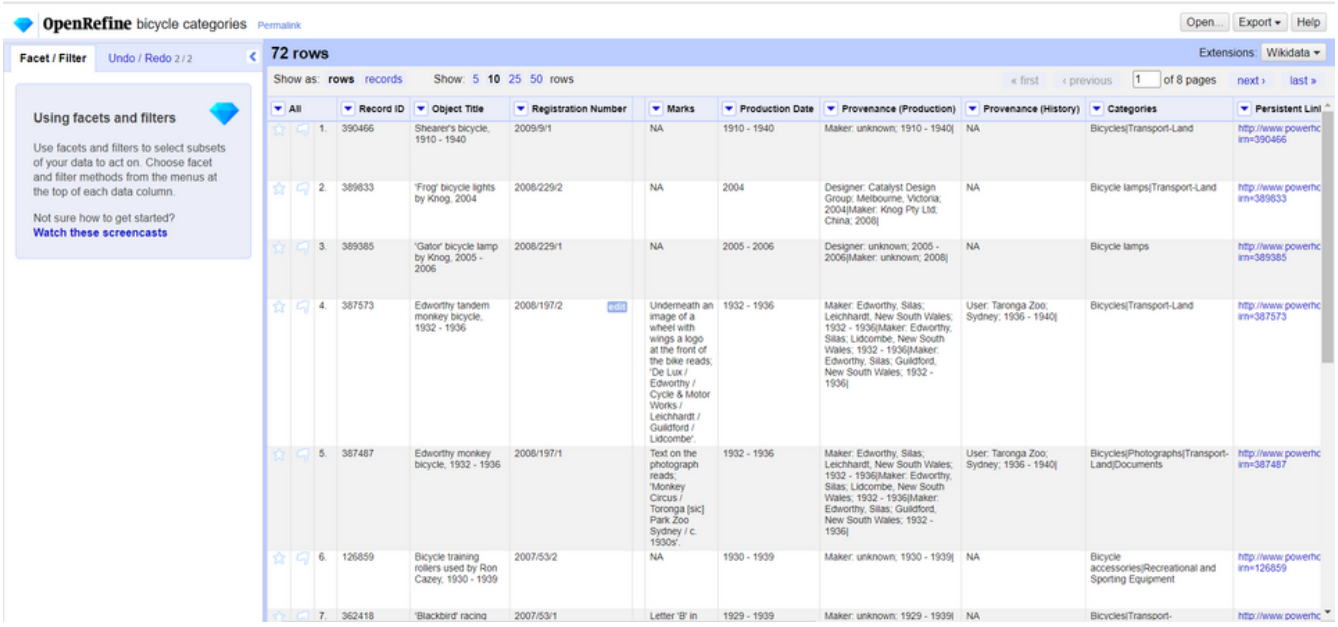


Abbildung 1. OpenRefine GUI

Dies sind einige wichtige Begriffe, die in OpenRefine verwendet werden: Da OpenRefine ein tabellarisches Datenmodell verwendet, gibt es **Columns** (ähnliche Begriffe: Feld, Attribute, Zellen) und **Rows** (ähnliche Begriffe: Datensätze, Zeilen). Dann gibt es noch die Option der **reconciliation** (ähnliche Begriffe: Integration, Zusammenführung), den Prozess des Abgleichs Ihres Datensatzes mit dem einer externen Quelle.

OpenRefine verfügt ausserdem über eine leistungsstarke **Undo/Redo** bzw. eine Historie-Funktion, sodass man ungehindert mit den Daten arbeiten und experimentieren kann, so viel man möchte. Jede Aktion, die Sie an einem Datensatz durchführen, wird von OpenRefine verfolgt und in diesem Fenster angezeigt, sodass man bequem alle gewünschten Transformationen an Ihrem Datensatz schnell und problemlos rückgängig machen kann.

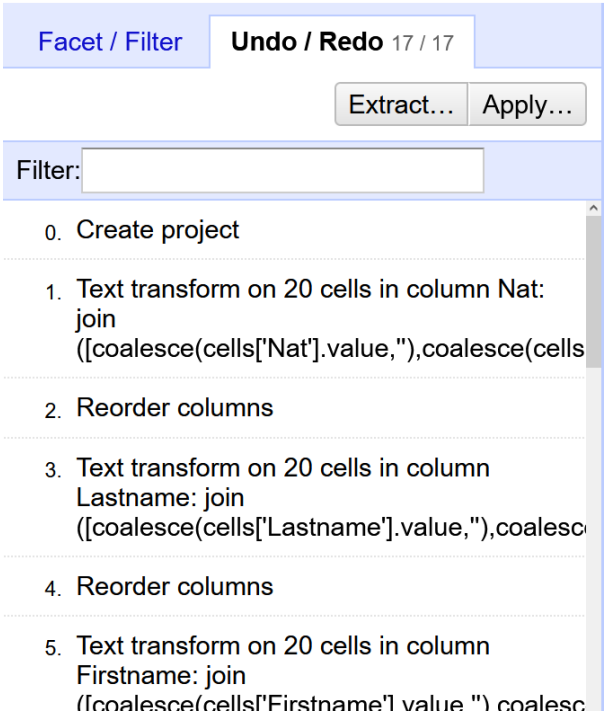
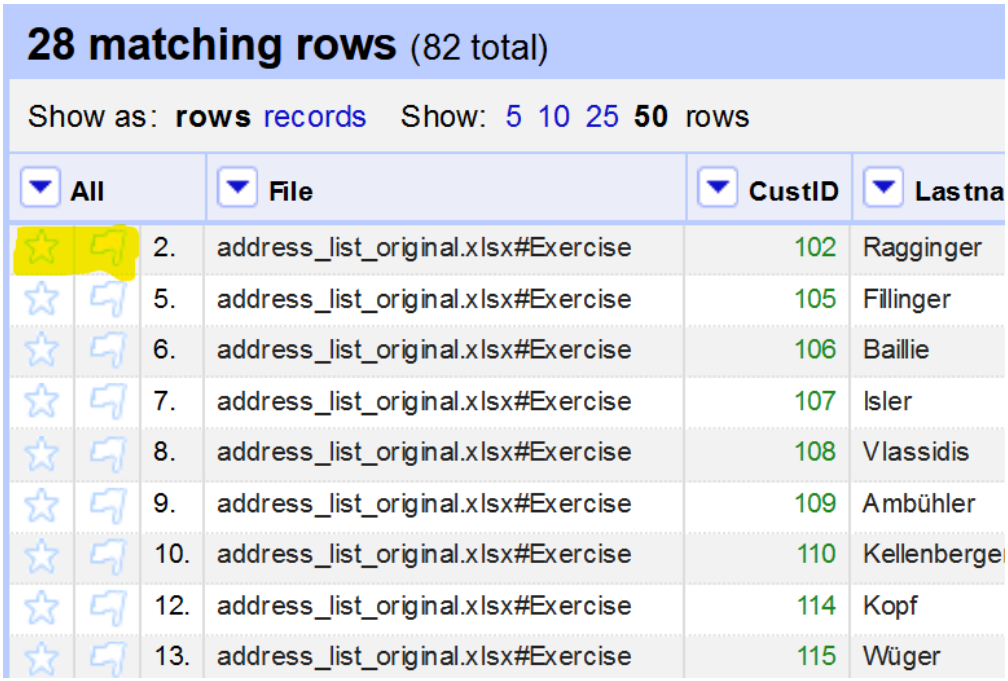


Abbildung 2. Undo/Redo Feature von OpenRefine.

Sie können auch fast alle vorgenommenen Transformations-Schritte im JSON-Format exportieren. Ausgenommen sind z.B. Ändernungen an einzelnen Zelleninahlten. Das exportierte File kann dann an einem späteren Zeitpunkt wieder importiert werden.

Auf der linken Seite jeder Zeile in Ihrem Datensatz befinden sich ein Flaggen- und ein Sternsymbol. Die Markierung bleibt bestehen, auch wenn Sie Ihr Projekt schliessen/öffnen. Der Sinn dieser Markierung ist, dass der Benutzer bestimmte Zeilen mit einem Stern oder einer Flagge versehen kann, um diese auch später wiederzufinden.

Sterne/Flaggen können einzeln oder zu allen passenden Zeilen hinzugefügt werden, indem Sie die spezielle Spalte *All* verwenden. Eine 'Facettierung' (die Sie später im Arbeitsblatt lernen werden) nach Sternen/Flags ist ebenfalls möglich.



28 matching rows (82 total)

Show as: **rows** records Show: 5 10 25 **50** rows

All		File	CustID	Lastna
	2.	address_list_original.xlsx#Exercise	102	Ragginger
	5.	address_list_original.xlsx#Exercise	105	Fillinger
	6.	address_list_original.xlsx#Exercise	106	Baillie
	7.	address_list_original.xlsx#Exercise	107	Isler
	8.	address_list_original.xlsx#Exercise	108	Vlassidis
	9.	address_list_original.xlsx#Exercise	109	Ambühler
	10.	address_list_original.xlsx#Exercise	110	Kellenberge
	12.	address_list_original.xlsx#Exercise	114	Kopf
	13.	address_list_original.xlsx#Exercise	115	Wüger

Abbildung 3. Die Sternen- und Flaggensymbole sind jeweils auf der linken Seite jeder Zeile zu finden

Projekte

Ein **Projekt** in OpenRefine steht jeweils für einen importierten Datensatz. Es wird erstellt, indem Sie bestehende Daten in OpenRefine importieren. Zuerst importieren Sie Ihre Daten (über einen der bereitgestellten Quellen), dann konfigurieren Sie die Parsing-Optionen und fahren mit der Erstellung des Projekts fort. Sobald Sie das Projekt erstellt haben, wird es in einer separaten Datei im OpenRefine-Verzeichnis gespeichert. Die ursprüngliche Datenquelle wird nach dem Erstellen des Projekts nicht mehr benötigt und wird auch nicht von OpenRefine verändert.

OpenRefine speichert die Daten in einem separaten Verzeichnis, nachdem das Projekt erstellt wurde, und verändert niemals die originale Datenquelle. Weitere Informationen darüber, wie OpenRefine mit Projekten umgeht und wie sie gespeichert werden, finden Sie in der OpenRefine-Dokumentation.

Das **Erkunden von Daten** in OpenRefine hilft Ihnen, mehr über Ihren Datensatz zu erfahren. Typische Fragen beim Erkunden sind:

- Welche Attribute sind Text, welche sind diskrete oder kontinuierliche Zahlen?
- Welche Art von Werten hat jedes Attribut?
- Wie sind die Werte verteilt?

In OpenRefine gibt es unterschiedliche Arten, die Daten darzustellen. Im Wesentlichen unterscheidet man zwischen **Rows** und **Records**. Mehr Infos dazu gibt es auf der OpenRefine Dokumentation

OpenRefine weist standardmässig jeder Zelle den Datentyp `string` zu. Nach Wunsch kann jedoch eine der folgenden Basistypen zugeordnet werden `number`, `boolean`, `date ISO-8601`. Diese Datentypen sind eine wichtige Voraussetzung, dass bestimmte Funktionen durchgeführt werden können z.B.

Summieren. Zugewiesene Datentypen werden durch eine grüne Schrift hervorgehoben.

Man beachte, dass jeder Zelle ein eigener Datentyp zugeordnet wird. In diesem Punkt ähnelt OpenRefine Excel mehr als einer Datenbank, da in OpenRefine nicht zwingend alle Zellen einer Spalte denselben Datentyp haben - obschon dies mit Blick auf Auswertungen (und Datenbank-Tabellen) unabdingbar ist. Die Lösung dazu bieten OpenRefine-Funktionen (sog. *Facetten*), die unten beschrieben und geübt werden.

4. OpenRefine-Funktionen

In diesem Abschnitt werden Sie zunächst den Reader/Input von OpenRefine kennenlernen und sich mit seinen Funktionen vertraut machen. Anschliessend werden Transformationsfunktionen thematisiert. Schlussendlich wird beschrieben, wie der Export von Daten in OpenRefine gehandhabt wird.

Eine der wertvollsten Funktionen von OpenRefine sind 'Facetten', die es ermöglichen, die Werte einer bestimmten Spalte schnell zu untersuchen, sowie Transformationen, die viele Optionen zur Manipulation der Daten bieten. Das Verstehen dieser Funktionen ist der Schlüssel zum Verständnis des gesamten "Lebenszyklus" eines OpenRefine-Projekts und wird auch in den Übungen auf den Arbeitsblättern von Nutzen sein.

Importieren und Erkunden von Daten

Um OpenRefine zu nutzen, muss zunächst eine Datei von Ihrer Maschine oder aus dem Internet importiert werden, aus der dann ein **Projekt** erstellt wird. Nach der Datenbereinigung/-manipulation können Sie die Daten dann in ein bestimmtes Dateiformat exportieren und verwenden.

Importieren

Das Importieren ist der erste Schritt beim Arbeiten mit OpenRefine. Die folgenden Formate werden für den Import in OpenRefine unterstützt: CSV, TSV, JSON, XML, Microsoft Excel-Tabellen (`.xslx`, `.xls`), HTML-Tabellen (`.html`), Google Spreadsheets (online).

OpenRefine bietet auch die Möglichkeit, Daten aus einer bestehenden Datenbank oder aus Services zu importieren.

Facetten

Nachdem man eine Datenquelle geöffnet hat, beginnt man normalerweise sich mit diesem Datensatz näher auseinanderzusetzen. Man will also den Dateninhalt "sehen" und auch die Datentypen der Spalten bestimmen. Datentypen sind wichtig, weil sie das Arbeiten mit den Daten erleichtern, indem sie einzigartige Operationen ermöglichen. Einige grundlegende Datentypen sind Text, Zahl, Datum/Zeit, Boolean (true/false) und Enumeration (rot, gelb, grün).

Datentypen sind wichtig. Programme wie Superset/Tableau/PowerBI und sogar MS Excel haben integrierte Funktionen, um den Typ zu erraten. In OpenRefine finden Sie die Option "Parse cell text into numbers, dates, ...", wenn Sie ein CSV importieren.

Facetten sind eine wichtige Funktion von OpenRefine, die über Tabellenkalkulationen und traditionelle Datenbankwerkzeuge hinausgeht. Sie zeigen die Datenvarianz in einer bestimmten Spalte an. Die Facettierung ermöglicht es uns, einen besseren Überblick über den gesamten Datensatz zu erhalten und die Daten aus einer grösseren Perspektive zu betrachten.

Eine Facette gruppiert Werte (z.B. vom Datentyp Text oder numerisch), die sich in einer Spalte befinden, und ermöglicht es dem Benutzer, die Werte in verschiedenen Zellen gleichzeitig zu filtern und zu bearbeiten, was vergleichbar mit einer Browsing-Funktion ist.

Normalerweise werden Facetten für eine bestimmte Spalte erstellt, indem Sie auf die Spalte klicken, die Option *Facette* auswählen und dann auf eine der Facettenalternativen klicken. Wenn Sie z.B. **[Text Facet]** auswählen, wird der gesamte Inhalt der Spaltenzellen verglichen, um allgemeine Informationen über die Werte dieser Spalte anzuzeigen. Sie können nun entweder selber einen neuen Wert für den Facetteneintrag eingeben oder die OpenRefine-Option *Clustering* verwenden, die sich automatisch um die Facette kümmert, indem sie ähnliche Einträge zusammenfasst.

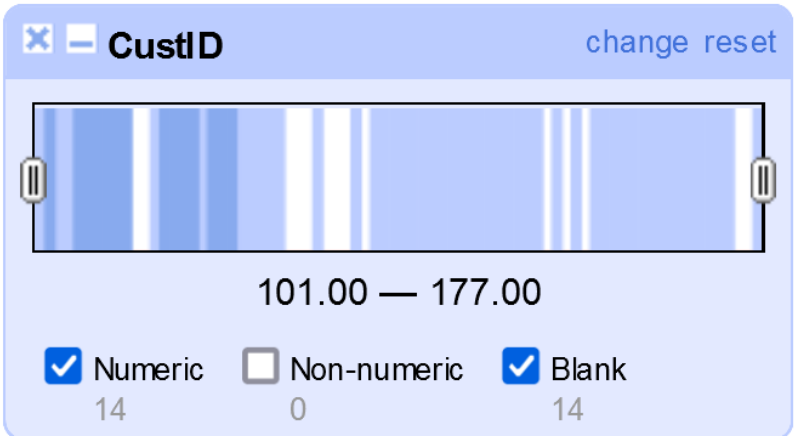


Abbildung 4. Beispiel für eine nummerische Fassette `customer_id`

Filtern

Es kann auch nach einem bestimmten Wert im Datensatz gefiltert werden, indem Sie die Option **[Textfilter]** im Dropdown-Menü der Spalte wählen. Dadurch wird ein Textfeld erstellt, in das Sie den Text einfügen können, den Sie zum Filtern der Spaltenwerte benötigen.

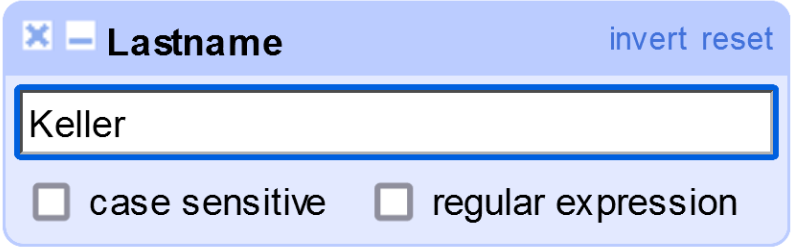


Abbildung 5. Beispiel für das Filtern nach einem Text in der Spalte `Lastname`

Clustering

Clustering ist eine weitere wichtige Funktion zur Gruppierung ähnlicher Daten, die Ihnen hilft, Dateninkonsistenzen und Rechtschreibfehler zu erkennen. Dies ist bei vielen Datensätzen üblich.

Das Clustering kann durch Auswahl einer *Facette* auf der Spalte, die Sie clustern möchten, und durch Auswahl der Methode, mit der Sie Ihre Spaltenwerte clustern möchten, erfolgen.

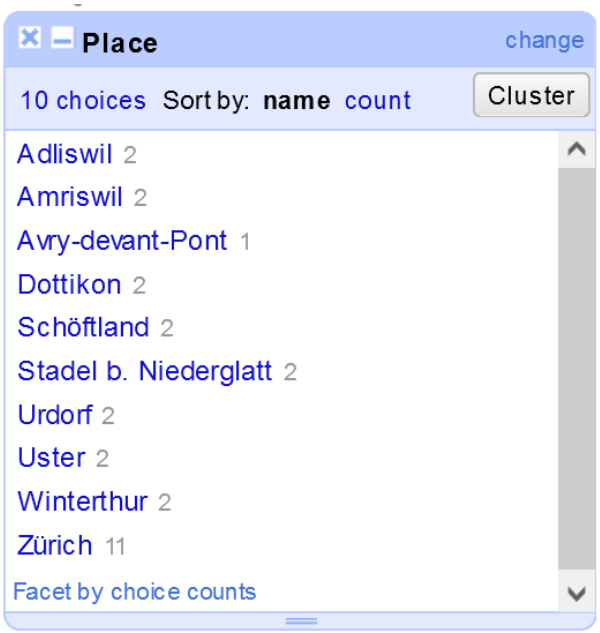


Abbildung 6. Der Cluster-Button in OpenRefine

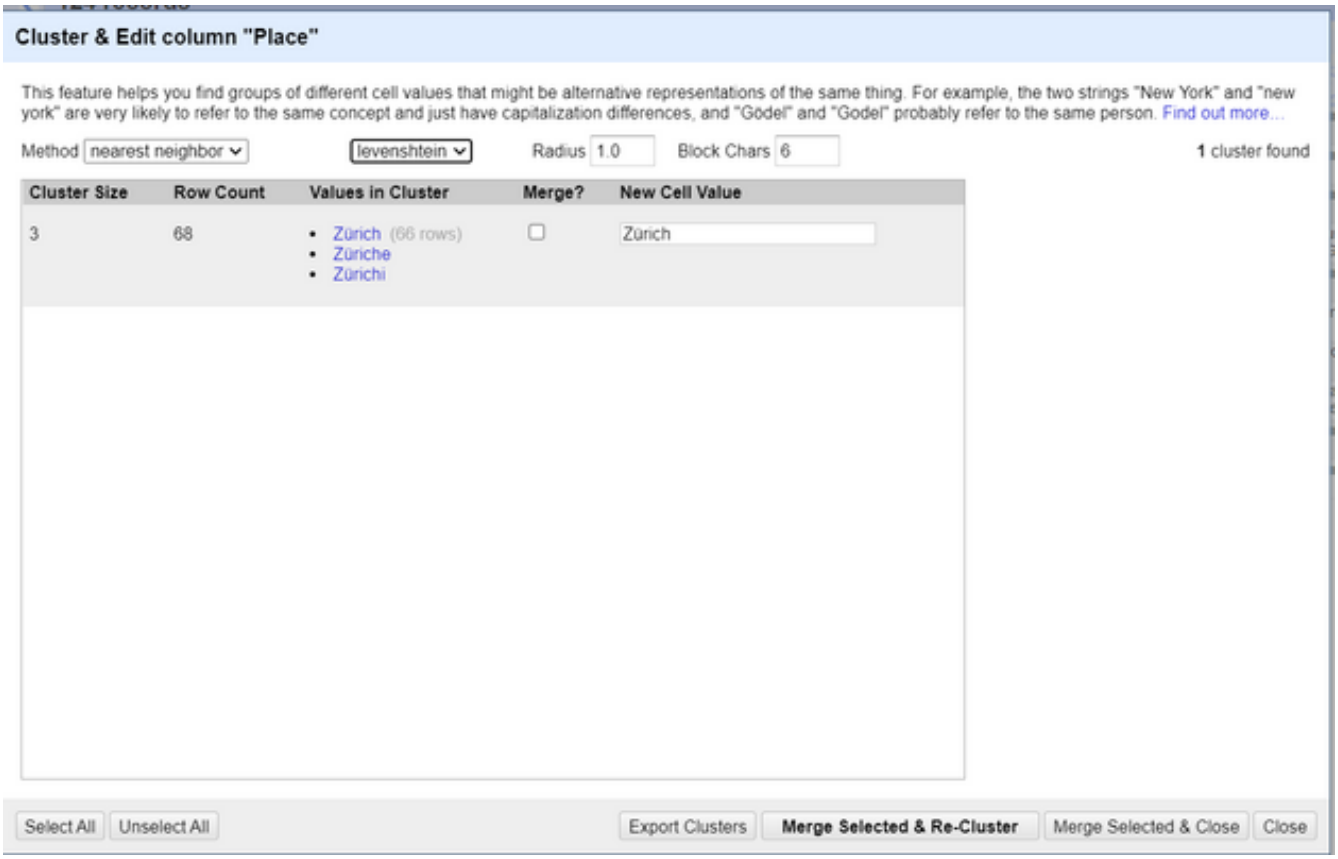


Abbildung 7. Beispiel für das Benutzen der "nearest neighbor" clustering Methode.

Wie im obigen Bild zu sehen ist, handelt es sich um ein Beispiel für die Stadt *Zürich*, bei dem der

Name der Stadt absichtlich ein paar Mal falsch geschrieben wurde. Die Clustering-Funktion findet diese Rechtschreibfehler und bietet uns Optionen für die Behandlung dieser Arten von Rechtschreibfehlern.

Weitere Informationen zum Clustering finden Sie auf dieser OpenRefine-Dokumentationsseite.

Transforming

OpenRefine bietet einige leistungsstarke Features/Funktionen für die Arbeit mit Daten. Einige Transformationsfunktionen umfassen unter anderem:

- Auf Wertebene:
  - String-Operationen
- Auf Feld-/Spaltenebene:
  - Teilen und Verbinden mehrwertiger Zellen
  - Berechnungen in Feldern
  - Hinzufügen von Konstanten
  - Verbinden (Verketten) von Feldern

Es gibt auch andere OpenRefine-Funktionen, mit denen Sie Ihren gesamten Datensatz mit nur wenigen Klicks umwandeln können (Massenbearbeitung), z.B.:

- Neuordnung von Spalten:
  - Einzelne Neuordnung - befindet sich im Dropdown-Menü der **Spalte › Edit Column › Move column to [direction]**.
  - Mehrfache Neuanordnung - befindet sich im Dropdown-Menü der Spalte **All › Edit Column › Re-order / remove columns...**
- Umbenennung von Spalten:
  - Befindet sich im Dropdown-Menü der **Spalte › Edit Column › Rename this column**.
- Daten sortieren:
  - Im Dropdown-Menü der Spalte **[Sort...]**, danach müssen Sie den Datentyp der Spaltenwerte auswählen.

String Operations und Funktionen

OpenRefine bietet einige schnelle Transformationsoperationen, die beim Umgang mit Daten nützlich sind. Sie können entweder eine der voreingestellten Transformationen verwenden (sofort einsatzbereit) oder die GREL-Sprache verwenden, um Ihre eigene Transformationsfunktion zu implementieren (erfordert leichte Programmierkenntnisse; siehe das separate Kapitel unten).

Einige der String-Operationen umfassen:

- Ersetzen von Anführungszeichen (häufig bei unsauberen Daten)
- Umwandlung von Text in andere Datentypen
- Schneiden von Leerzeichen oder anderen Sonderzeichen
- Escapen/Unterschneiden von HTML-Zeichen

Wenn man diese Operationen richtig einsetzt, kann man unordentliche Daten schnell in saubere und maschinenlesbare Informationen umwandeln.

Die Verwendung von *String-Operationen* in OpenRefine erfolgt durch Klicken auf die Option **[Edit Cells]** im Dropdown-Menü der Spalte. Bei der Verwendung von String-Operationen können Sie eine der voreingestellten Transformationen wählen, die von OpenRefine angeboten werden, oder eigene Transformationen mit der GREL-Sprache von OpenRefine schreiben, z.B. den Ausdruck `value.toDate()` (siehe Abschnitt unten).

Felder aufteilen

Eine weitere wichtige Funktion von OpenRefine ist die *Split*-Funktion, die für die Aufteilung von Spaltenwerten in mehrere Spalten nützlich ist. Alles, was Sie tun müssen, ist, das Zeichen zwischen den Wörtern anzugeben, und dann teilt OpenRefine diese Zeichenfolgen für Sie in mehrere Spalten auf. Dies ist sehr häufig der Fall, wenn Sie mit unübersichtlichen Daten arbeiten, denn viele Daten in einer Spalte machen viel mehr Sinn, wenn sie in mehrere Spalten aufgeteilt werden.

Das *Aufteilen von Feldern* in OpenRefine erfolgt durch Klicken auf **[Edit Column]** und dann **[Split into several Columns...]**. Hier können Sie ein oder mehrere Trennzeichen oder eine Feldlänge für die Trennung Ihrer Spaltenwerte angeben.

Sie können auch **mehrwertige Zellen** (mit zusätzlichen Optionen) aufteilen, indem Sie die Option **[Split multi-valued Cells...]** verwenden.

Felder verknüpfen

Eine weitere Option, die gewissermassen das Gegenteil der Aufteilung von Feldern ist, heisst **[Joining Fields]** und wird verwendet, wenn Felder mit einem Trennzeichen (oder ohne) verbunden werden. Das Wort ist ziemlich selbsterklärend.

Das *Verbinden von Feldern* in OpenRefine erfolgt durch Klicken auf **[Edit Column]** und dann **[Join Columns...]**. Daraufhin werden Ihnen die zu verbindenden Spalten und das Trennzeichen zwischen den Inhalten der einzelnen Spalten zur Auswahl angeboten. Sie können die Ergebnisse in dieselbe Spalte schreiben oder spontan eine neue Spalte erstellen.

Sie können auch **mehrwertige Zellen** verbinden (genau wie bei der obigen Aufteilung), indem Sie die Option **[Join multi-valued Cells...]** verwenden.

Zusammenführung von Datensätzen aus verschiedenen Quellen

Ein häufiges Szenario bei der Arbeit mit Daten ist der Bezug von Daten aus mehreren Quellen. Auch wenn die eingehenden Daten letztlich einem gemeinsamen Zweck dienen, ist die Struktur der Daten in den verschiedenen Quellen möglicherweise nicht identisch. Ein Beispiel hierfür wäre der Erhalt zweier Kundenlisten mit denselben Attributen, aber z.B. mit unterschiedlichen Spaltennamen. Obwohl es aus menschlicher Sicht so aussieht, als wären es dieselben Daten, ist das für Computer nicht der Fall. Dabei müssen verschiedene Techniken angewendet werden, um diese verschiedenen Quellen ähnlicher Daten zu integrieren oder besser gesagt zu "verschmelzen".

Beim Zusammenführen von Datensätzen gibt es in der Regel drei Arten von Fällen:

1. Erweitern des ersten Datensatzes um einen weiteren, der meist überlappende Spaltennamen hat, auch bekannt als **vertikal nach unten erweitern**.
2. Anreicherung des ersten Datensatzes um nur eine oder wenige Spalten eines anderen Hilfsdatensatzes, auch bekannt als **horizontale Erweiterung um eine oder wenige Spalten**
3. Erweitern des ersten Datensatzes und seiner Zeilen durch einen anderen ergänzenden Datensatz, ähnlich dem *SQL JOIN*, auch bekannt als **horizontal erweitern**.

In den Fällen 2 und 3 wird eine gemeinsame Spalte mit der Bezeichnung **Key** (Schlüssel) benötigt. Typische Schlüssel sind Identifikatoren wie Postleitzahlen, Postadressen (siehe Abschnitt Geokodierung weiter unten auf dem Arbeitsblatt), Ländercodes, Gemeindecodes, ISBN usw.

Einige der oben genannten Fälle werden in den Aufgaben 2 und 3 angewendet. Der erste Fall wird in Übung 2 und der zweite Fall in Übung 3 zur Anwendung kommen. Für den dritten Fall gibt es keine Übung auf diesem Arbeitsblatt, die ihn demonstriert. OpenRefine bietet derzeit keine direkte Lösung für diesen Fall.

Validierung und Deduplizierung

Validierung des Datensatzes

OpenRefine bietet auch Funktionen zur Validierung Ihrer Daten gegen einen anderen Datensatz. Dies kann durch die Erstellung einer *Benutzerdefinierten Textfacette* unter Verwendung der GREL-Funktion `cell.cross()` erfolgen, die Werte aus zwei verschiedenen Spalten vergleicht. Es gibt keinen nativen/geradlinigen Weg für die Validierung von Datensätzen, aber es kann durch die Verwendung von Funktionen erreicht werden, die sich auf die gleiche Weise wie die GREL-Funktion `cell.cross()` verhalten.

Daten-Deduplizierung

Um die Daten zu entduplizieren, kann man beispielsweise das **Clustering** nach Schlüsselkollisionen verwenden, um ähnliche Daten zu finden. Anschliessend kann man die doppelten Daten manuell bewerten und entscheiden, was mit ihnen geschehen soll, nachdem der Clustering-Prozess abgeschlossen ist. Dies ist in der Regel eine recht unkomplizierte und effiziente Methode zur Deduplizierung eines Datensatzes.

Exportieren

Wenn Sie die Arbeit mit den Daten abgeschlossen haben und die Daten im gewünschten Format vorliegen, ist der letzte Schritt das Exportieren dieser Daten in ein bestimmtes Dateiformat.

Die folgenden Formate werden für den Export in OpenRefine unterstützt: CSV, TSV, HTML-Tabelle, Microsoft Excel-Kalkulationstabelle (`.xlsx`, `.xls`), ODF-Kalkulationstabelle (`.ods`) und einige andere Export-Optionen.

Übung 1: Ein erster OpenRefine-Workflow

In dieser Übung wird einen typischen Arbeitsablauf und den Lebenszyklus eines OpenRefine-Projekts demonstiert, von der Erstellung bis zur Fertigstellung der Daten für den Export.

Daten

Für diese Aufgabe werden die Daten aus der Datei `address_list_original.xlsx` verwendet. Diese ist Teil des Zip-Archivs `Daten_OpenRefine.zip`, das Sie von dieser Seite herunterladen können. (Gleicher Abschnitt wie dieses Arbeitsblatt.)

Schritt 1: Projekt erstellen

Um ein Projekt (den eigentlichen Arbeitsbereich, in dem mit den Daten gearbeitet wird) zu erstellen, müssen Sie Folgendes tun:

1. Öffnen Sie OpenRefine in Ihrem lokalen Webbrowser.
2. Wählen Sie die Datei aus, mit der Sie arbeiten möchten (oder rufen Sie sie direkt aus dem Internet ab), in unserem Fall die Datei, welche Sie soeben über den obigen Link heruntergeladen haben.
3. Auf **[Weiter]** klicken.
4. Nachdem die zu importierende Datei ausgewählt ist, muss das Parsing konfiguriert werden. Die "bevorzugten" Optionen werden automatisch von OpenRefine ausgewählt, aber sie können nach Belieben angepasst werden. (siehe Abbildung 8 unten).
5. Dem Projekt einen Namen geben und auf **[Create Project]** klicken, um das Projekt in OpenRefine zu erstellen und mit der Arbeit daran zu beginnen (auch zur späteren Verwendung gespeichert).

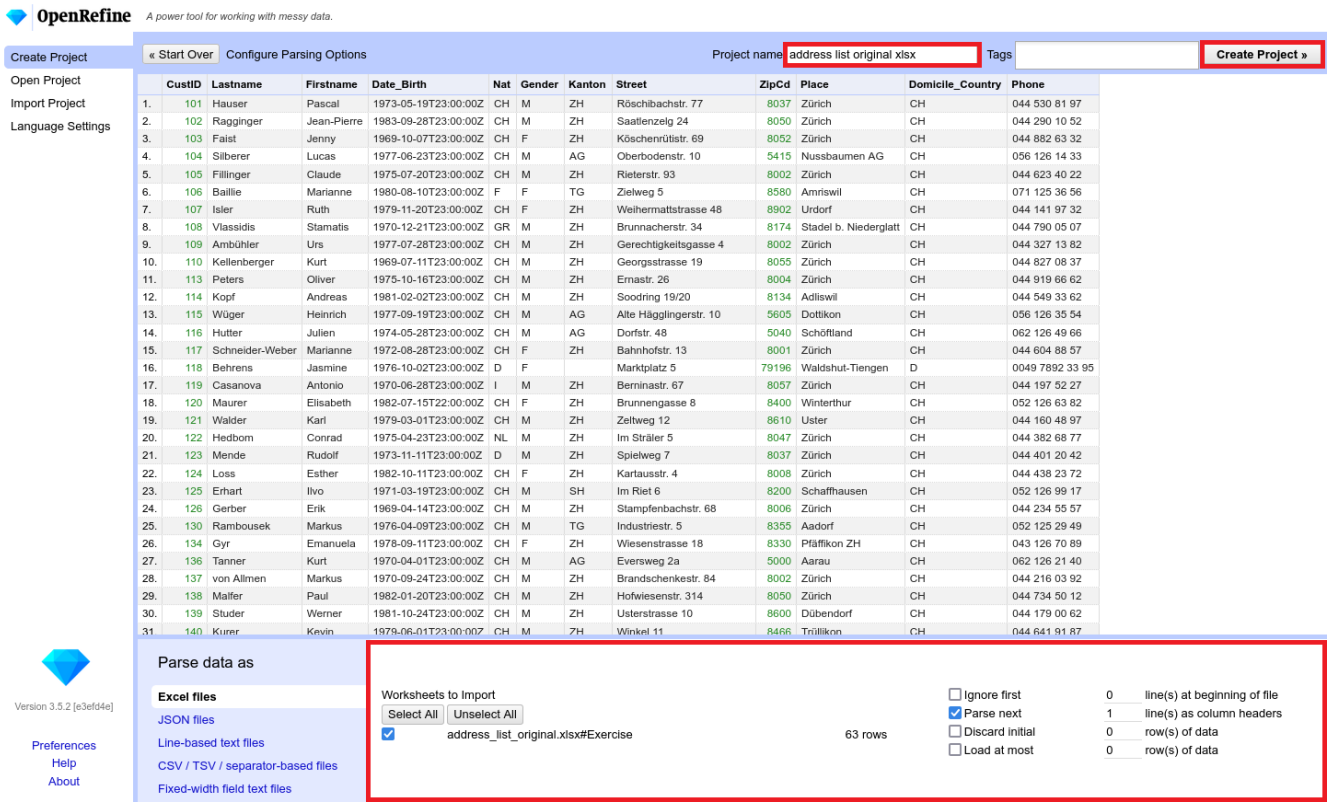


Abbildung 8. Projekt Parsing-Optionen.

Schritt 2: Daten überprüfen

Filtern Sie den Datensatz, um nur Kunden aus dem Kanton Zürich herauszufiltern und deren Adressen in einer neuen Spalte mit der vollständigen Adresse zusammenzuführen.

1. Erstellen Sie eine Textfacette auf der Spalte `Kanton`, um nur Kunden zu filtern, die im Kanton `ZH` wohnen.
2. Verschmelzen Sie die Spalten `Street`, `Place` und `ZipCd` zu einer neuen Spalte, getrennt durch das Zeichen `,` (ausser der Spalte `ZipCd`).
3. Fügen Sie die Landesvorwahl zu den Telefonnummern hinzu.
   a. Fügen Sie `+41` als Ländervorwahl hinzu (für die Schweiz)
   b. Entfernen Sie die Vorwahl `0` aus den Telefonnummern
   c. Entfernen Sie die Leerzeichen aus der Telefonnummer (z.B. `+41445308197`)

Eine Möglichkeit zur Manipulation von Zeichenketten (dritter Schritt) besteht in der Anwendung des folgenden GREL-Ausdrucks: `value.replace(value, "+41" + value.substring(1)).replace(" ", "")`. Es gibt selbstverständlich noch andere Möglichkeiten.

Schritt 3: Daten Exportieren

Nachdem Sie die oben genannten Aufgaben erledigt haben, können Sie Ihren Datensatz (Projekt) exportieren, indem Sie auf **Export › Excel 2007+** klicken. OpenRefine exportiert dann Ihren aktuellen Datensatz in eine neue Excel-Tabelle.

Übung 2: Integrieren eines anderen Datensatzes

In dieser Übung werden Sie einen Quell-Datensatz in einen vorgegebenen Ziel-Datensatz integrieren. Der Quelldatensatz enthält ähnliche Daten wie der Zieldatensatz, jedoch mit unterschiedlichen Spaltennamen und Strukturen. Solche Szenarien kommen im praktischen Umfeld relativ oft vor, da es eine Vielzahl von Datenquellen mit praktisch denselben Daten gibt. Letztendlich müssen alle diese Datenquellen, in einem endgültigen Datensatz zusammengeführt werden, der alle Daten aus allen Quellen enthält.

Sie werden verschiedene Methoden und Techniken in OpenRefine anwenden müssen, um zwei Datensätze erfolgreich in einen einzigen zusammenzuführen. Dazu gehören das Aufteilen/Zusammenführen von Feldern, das Mapping verschiedener Attribute, die Deduplizierung der Daten usw.

Diese Übung dient auch zur Veranschaulichung des ersten Falls (*vertikal nach unten erweitern*) des Kapitels *Datensätze aus verschiedenen Quellen verbinden*, das Sie zuvor im Arbeitsblatt gelesen haben.

Daten

Für diese Aufgabe werden die Daten aus den Files `address_list_original.xlsx` & `address_list_scrambled.xlsx` verwendet.

Schritt 1: Projekt erstellen

Wenn Sie die beiden Excel-Tabellen heruntergeladen haben, erstellen Sie das Projekt, indem Sie:

- Hochladen der beiden Dateien als Projektdateien (OpenRefine unterstützt den Import mehrerer Dateien).
- Wählen Sie beide Dateien auf der Seite **[Zu importierende Dateien auswählen]**.
- Belassen Sie die Standard-Parsing-Optionen und klicken Sie auf **[Projekt erstellen]**.

Schritt 2: Integration der Daten

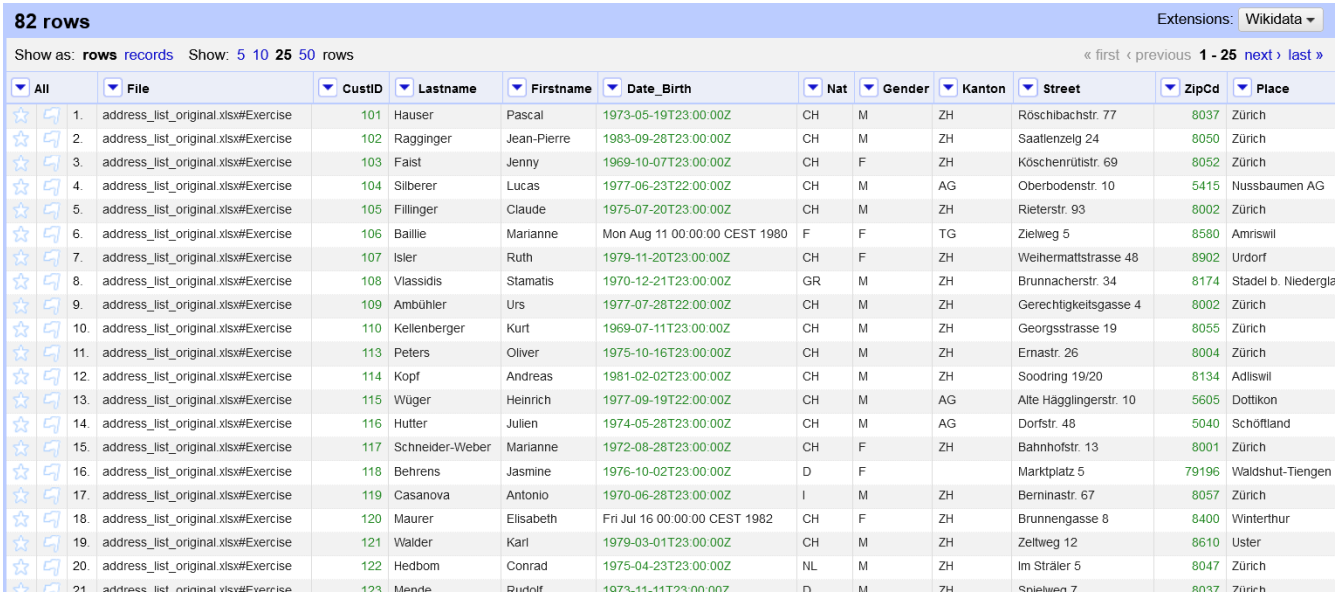
Der *Zieldatensatz* ist derselbe wie in der ersten Übung, und hier geht es darum, einen anderen Datensatz, d. h. den *Quelldatensatz* (mit ähnlichen Spaltennamen und -werten) in den *Zieldatensatz* zu integrieren, einschliesslich korrektem Mapping und erfolgreicher Datenintegration und Deduplizierung.

Nachdem Sie das Projekt erstellt haben, sehen Sie eine neue Spalte mit der Bezeichnung "File", die der Datei entspricht, zu der der Datensatz gehört. Diese Spalte "File" wird auch als **Key** (Schlüssel)-Spalte bezeichnet, wenn zwei Datensätze zusammengeführt werden, da sie die Spalte ist, auf die sich unser Datensatz stützt, um zu unterscheiden, woher die Daten stammen (siehe auch den ersten Fall im Kapitel *Datensätze aus verschiedenen Quellen zusammenführen* weiter oben im Arbeitsblatt).

Der Datensatz zeigt eine Summe aller Zeilen aus beiden Dateien, die alle Spalten aus beiden Dateien enthalten. Nachdem die Datensätze aus der ersten Datei (Original-/Zieldatensatz) zu Ende sind, beginnt die Anzeige der Datensätze aus der zweiten Datei (verschlüsselter/Quellendatensatz). Sie

werden auch feststellen, dass einige Spalten (absichtlich) denselben Namen haben und daher jeder Datensatz Daten zu diesen Spalten enthält.

Nachfolgend finden Sie einige Screenshots, die zeigen, wie die Daten aussehen und wie sich die Datensätze von der ersten zur zweiten Datei entwickeln.



82 rows

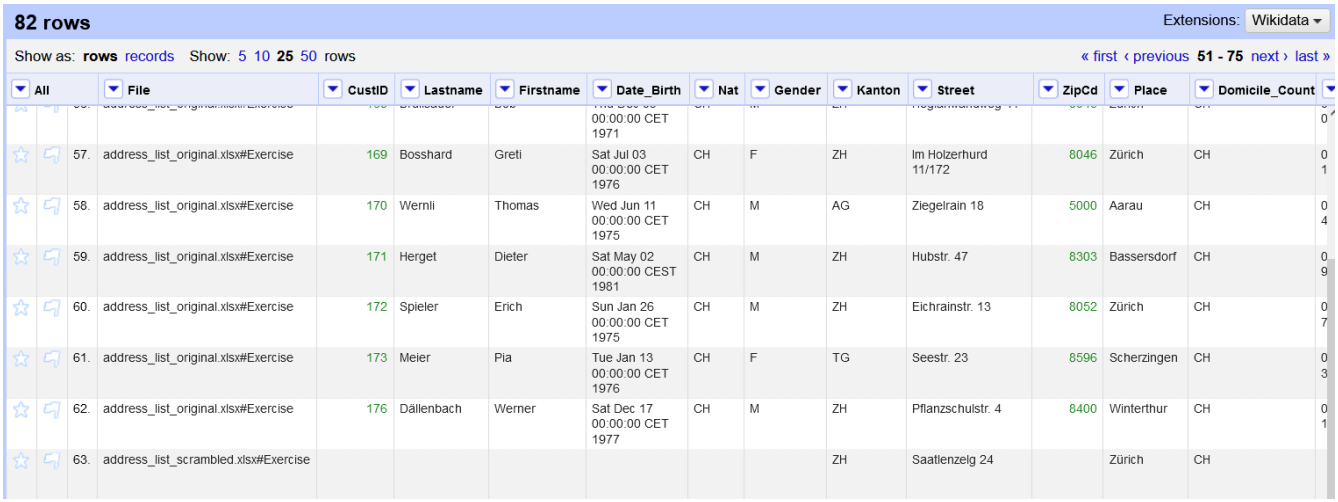
Extensions: Wikidata

Show as: **rows** records Show: 5 10 **25** 50 rows

« first ‹ previous **1 - 25** next › last »

	All	File	CustID	Lastname	Firstname	Date_Birth	Nat	Gender	Kanton	Street	ZipCd	Place
1.		address_list_original.xlsx#Exercise	101	Hauser	Pascal	1973-05-19T23:00:00Z	CH	M	ZH	Röschibachstr. 77	8037	Zürich
2.		address_list_original.xlsx#Exercise	102	Ragginger	Jean-Pierre	1983-09-28T23:00:00Z	CH	M	ZH	Saatlenzelg 24	8050	Zürich
3.		address_list_original.xlsx#Exercise	103	Faist	Jenny	1969-10-07T23:00:00Z	CH	F	ZH	Köschenrütistr. 69	8052	Zürich
4.		address_list_original.xlsx#Exercise	104	Silberer	Lucas	1977-06-23T22:00:00Z	CH	M	AG	Oberbodenstr. 10	5415	Nussbaumen AG
5.		address_list_original.xlsx#Exercise	105	Fillinger	Claude	1975-07-20T23:00:00Z	CH	M	ZH	Rieterstr. 93	8002	Zürich
6.		address_list_original.xlsx#Exercise	106	Baillie	Marianne	Mon Aug 11 00:00:00 CEST 1980	F	F	TG	Zielweg 5	8580	Amriswil
7.		address_list_original.xlsx#Exercise	107	Isler	Ruth	1979-11-20T23:00:00Z	CH	F	ZH	Weihermattstrasse 48	8902	Urdorf
8.		address_list_original.xlsx#Exercise	108	Vlassidis	Stamatis	1970-12-21T23:00:00Z	GR	M	ZH	Brunnacherstr. 34	8174	Stadel b. Niedergla
9.		address_list_original.xlsx#Exercise	109	Ambühler	Urs	1977-07-28T22:00:00Z	CH	M	ZH	Gerechtigkeitsgasse 4	8002	Zürich
10.		address_list_original.xlsx#Exercise	110	Kellenberger	Kurt	1969-07-11T23:00:00Z	CH	M	ZH	Georgsstrasse 19	8055	Zürich
11.		address_list_original.xlsx#Exercise	113	Peters	Oliver	1975-10-16T23:00:00Z	CH	M	ZH	Ernastr. 26	8004	Zürich
12.		address_list_original.xlsx#Exercise	114	Kopf	Andreas	1981-02-02T23:00:00Z	CH	M	ZH	Soodring 19/20	8134	Adliswil
13.		address_list_original.xlsx#Exercise	115	Wüger	Heinrich	1977-09-19T22:00:00Z	CH	M	AG	Alte Hägglingerstr. 10	5605	Dottikon
14.		address_list_original.xlsx#Exercise	116	Hutter	Julien	1974-05-28T23:00:00Z	CH	M	AG	Dorfstr. 48	5040	Schöftland
15.		address_list_original.xlsx#Exercise	117	Schneider-Weber	Marianne	1972-08-28T23:00:00Z	CH	F	ZH	Bahnhofstr. 13	8001	Zürich
16.		address_list_original.xlsx#Exercise	118	Behrens	Jasmine	1976-10-02T23:00:00Z	D	F		Marktplatz 5	79196	Waldshut-Tiengen
17.		address_list_original.xlsx#Exercise	119	Casanova	Antonio	1970-06-28T23:00:00Z	I	M	ZH	Berninastr. 67	8057	Zürich
18.		address_list_original.xlsx#Exercise	120	Maurer	Elisabeth	Fri Jul 16 00:00:00 CEST 1982	CH	F	ZH	Brunnengasse 8	8400	Winterthur
19.		address_list_original.xlsx#Exercise	121	Walder	Karl	1979-03-01T23:00:00Z	CH	M	ZH	Zeltweg 12	8610	Uster
20.		address_list_original.xlsx#Exercise	122	Hedbom	Conrad	1975-04-23T23:00:00Z	NL	M	ZH	Im Sträler 5	8047	Zürich
21.		address_list_original.xlsx#Exercise	123	Mende	Rudolf	1973-11-11T23:00:00Z	D	M	ZH	Spielweg 7	8037	Zürich

Abbildung 9. Daten in der Tabelle



82 rows

Extensions: Wikidata

Show as: **rows** records Show: 5 10 **25** 50 rows

« first ‹ previous **51 - 75** next › last »

	All	File	CustID	Lastname	Firstname	Date_Birth	Nat	Gender	Kanton	Street	ZipCd	Place	Domicile_Count
						00:00:00 CET 1971							0
57.		address_list_original.xlsx#Exercise	169	Bosshard	Greti	Sat Jul 03 00:00:00 CET 1976	CH	F	ZH	Im Holzerhurd 11/172	8046	Zürich	CH 0 1
58.		address_list_original.xlsx#Exercise	170	Wernli	Thomas	Wed Jun 11 00:00:00 CET 1975	CH	M	AG	Ziegelrain 18	5000	Aarau	CH 0 4
59.		address_list_original.xlsx#Exercise	171	Herget	Dieter	Sat May 02 00:00:00 CEST 1981	CH	M	ZH	Hubstr. 47	8303	Bassersdorf	CH 0 9
60.		address_list_original.xlsx#Exercise	172	Spieler	Erich	Sun Jan 26 00:00:00 CET 1975	CH	M	ZH	Eichrainstr. 13	8052	Zürich	CH 0 7
61.		address_list_original.xlsx#Exercise	173	Meier	Pia	Tue Jan 13 00:00:00 CET 1976	CH	F	TG	Seestr. 23	8596	Scherzingen	CH 0 3
62.		address_list_original.xlsx#Exercise	176	Dällenbach	Werner	Sat Dec 17 00:00:00 CET 1977	CH	M	ZH	Pflanzschulstr. 4	8400	Winterthur	CH 0 1
63.		address_list_scrambled.xlsx#Exercise							ZH	Saatlenzelg 24		Zürich	CH

Abbildung 10. Neue Daten aus der zweiten Datei

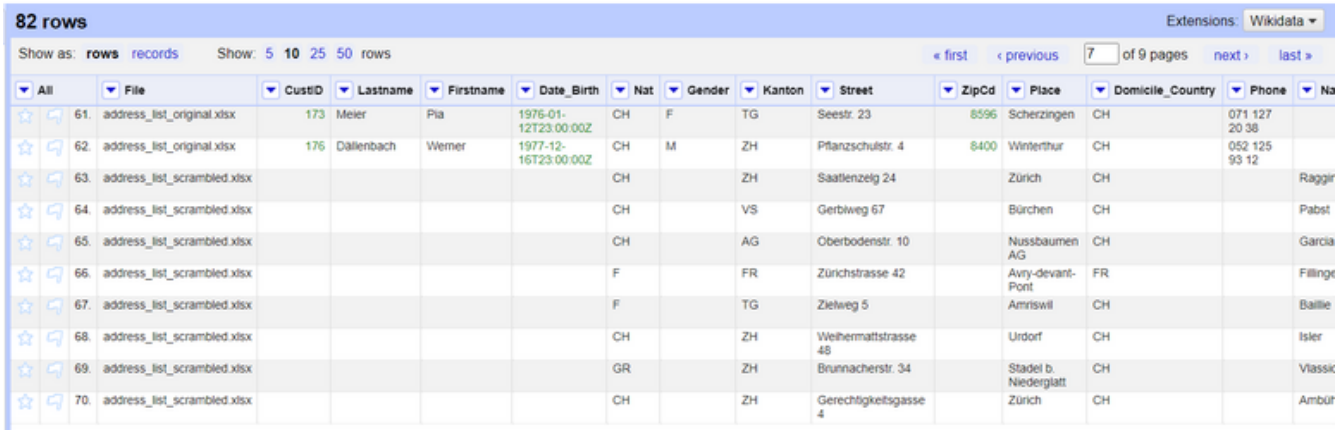
Schritt 2.1 Zusammenführung und Standardisierung der Spalten aus beiden Dateien

Zunächst müssen Sie die Spaltennamen (aus beiden Dateien) überprüfen und herausfinden, welche von ihnen zueinander gehören. Dieser Teil muss manuell durchgeführt werden.

Wenn Sie herausgefunden haben, dass die beiden Spalten der gleichen Sache entsprechen, müssen Sie diese zusammenführen. Dadurch werden alle Werte aus den beiden Dateien in einer einzigen Spalte zusammengefasst. Achten Sie darauf, dass Sie den *separator* leer lassen, wenn Sie die Spalten verbinden.

Sie werden sehen, dass in den verbundenen Spalten die erste Spalte auch die Werte aus der zweiten Spalte übernimmt und Sie einen Wert für jede unserer Zeilen (aus beiden Dateien) haben

werden.



82 rows

Extensions: Wikidata

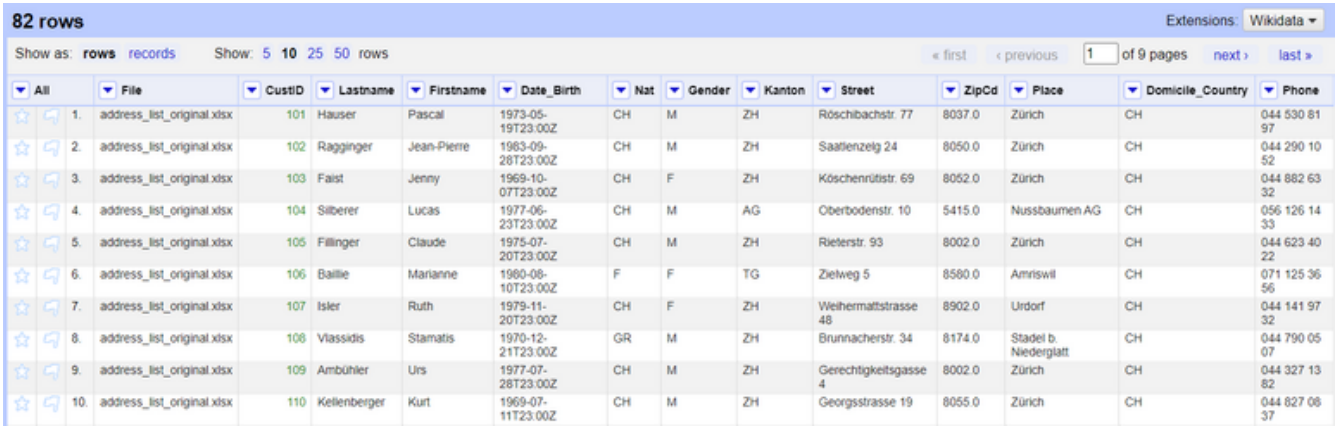
Show as: **rows** records Show: 5 **10** 25 50 rows « first ‹ previous 7 of 9 pages next › last »

All		File	CustID	Lastname	Firstname	Date_Birth	Nat	Gender	Kanton	Street	ZipCd	Place	Domicile_Country	Phone	Na
61.		address_list_original.xlsx	173	Meier	Pia	1976-01-12T23:00:00Z	CH	F	TG	Seestr. 23	8596	Scherzingen	CH	071 127 20 38	
62.		address_list_original.xlsx	176	Dällenbach	Werner	1977-12-16T23:00:00Z	CH	M	ZH	Pflanzschulstr. 4	8400	Winterthur	CH	052 125 93 12	
63.		address_list_scrambled.xlsx					CH		ZH	Saatlenzelg 24		Zürich	CH		Raggir
64.		address_list_scrambled.xlsx					CH		VS	Gerbiweg 67		Bürchen	CH		Pabst
65.		address_list_scrambled.xlsx					CH		AG	Oberbodenstr. 10		Nussbaumen AG	CH		Garcia
66.		address_list_scrambled.xlsx					F		FR	Zürichstrasse 42		Avry-devant-Pont	FR		Fillinge
67.		address_list_scrambled.xlsx					F		TG	Zielweg 5		Amriswil	CH		Baillie
68.		address_list_scrambled.xlsx					CH		ZH	Weihermattstrasse 48		Urdorf	CH		Isler
69.		address_list_scrambled.xlsx					GR		ZH	Brunnacherstr. 34		Stadel b. Niederglatt	CH		Vlassic
70.		address_list_scrambled.xlsx					CH		ZH	Gerechtigkeitsgasse 4		Zürich	CH		Ambüh

Abbildung 11. Die Spalten "Nat" und "Nationality" werden verbunden.

Führen Sie nun das Gleiche für alle übrigen Spalten durch und Sie erhalten einen Datensatz mit konsistenten Spalten.

Wenn Sie alle richtigen Spalten verbunden haben, entfernen Sie die anderen Spalten aus der zweiten Datei. Auf diese Weise sollten Sie einen Datensatz erhalten, der alle Daten aus beiden Dateien enthält, jedoch mit einheitlichen und standardisierten Spaltennamen (einige Spalten haben in beiden Dateien denselben Namen haben).



82 rows

Extensions: Wikidata

Show as: **rows** records Show: 5 **10** 25 50 rows « first ‹ previous 1 of 9 pages next › last »

All		File	CustID	Lastname	Firstname	Date_Birth	Nat	Gender	Kanton	Street	ZipCd	Place	Domicile_Country	Phone
1.		address_list_original.xlsx	101	Hauser	Pascal	1973-05-19T23:00Z	CH	M	ZH	Röschibachstr. 77	8037.0	Zürich	CH	044 530 81 97
2.		address_list_original.xlsx	102	Ragginger	Jean-Pierre	1983-09-28T23:00Z	CH	M	ZH	Saatlenzelg 24	8050.0	Zürich	CH	044 290 10 52
3.		address_list_original.xlsx	103	Faist	Jenny	1969-10-07T23:00Z	CH	F	ZH	Köschenrütistr. 69	8052.0	Zürich	CH	044 882 63 32
4.		address_list_original.xlsx	104	Silberer	Lucas	1977-06-23T23:00Z	CH	M	AG	Oberbodenstr. 10	5415.0	Nussbaumen AG	CH	056 126 14 33
5.		address_list_original.xlsx	105	Fillinger	Claude	1975-07-20T23:00Z	CH	M	ZH	Rieterstr. 93	8002.0	Zürich	CH	044 623 40 22
6.		address_list_original.xlsx	106	Baillie	Marianne	1960-08-10T23:00Z	F	F	TG	Zielweg 5	8580.0	Amriswil	CH	071 125 36 56
7.		address_list_original.xlsx	107	Isler	Ruth	1979-11-20T23:00Z	CH	F	ZH	Weihermattstrasse 48	8902.0	Urdorf	CH	044 141 97 32
8.		address_list_original.xlsx	108	Vlassidis	Stamatis	1970-12-21T23:00Z	GR	M	ZH	Brunnacherstr. 34	8174.0	Stadel b. Niederglatt	CH	044 790 05 07
9.		address_list_original.xlsx	109	Ambühler	Urs	1977-07-28T23:00Z	CH	M	ZH	Gerechtigkeitsgasse 4	8002.0	Zürich	CH	044 327 13 82
10.		address_list_original.xlsx	110	Kellenberger	Kurt	1969-07-11T23:00Z	CH	M	ZH	Georgsstrasse 19	8055.0	Zürich	CH	044 827 08 37

Abbildung 12. Das Datenset nachdem die Spalten verbunden wurden.

Schritt 2.2 Deduplizierung der Daten

Der nächste Schritt besteht darin, die Daten zu entduplizieren. Einige Daten aus der zweiten Datei sind auch in der ersten Datei vorhanden, was bedeutet, dass es sich um Duplikate handelt, die Sie entfernen und nur einen der Einträge behalten müssen. Es gibt auch andere Daten in der zweiten Datei, die in der ersten Datei nicht vorhanden sind (d.h. keine Duplikate), sodass Sie diese nicht entfernen müssen.

Damit ein Datensatz als doppelt gilt, muss die folgende Bedingung erfüllt sein: Wenn der Vorname, der Nachname und das Geburtsdatum identisch sind, handelt es sich um dieselbe Person, d.h. um ein Duplikat.

Dazu müssen Sie die Funktion *Duplikate-Facette* verwenden, die sich unter **Facet › Customized facets › Duplicates facet** befindet.

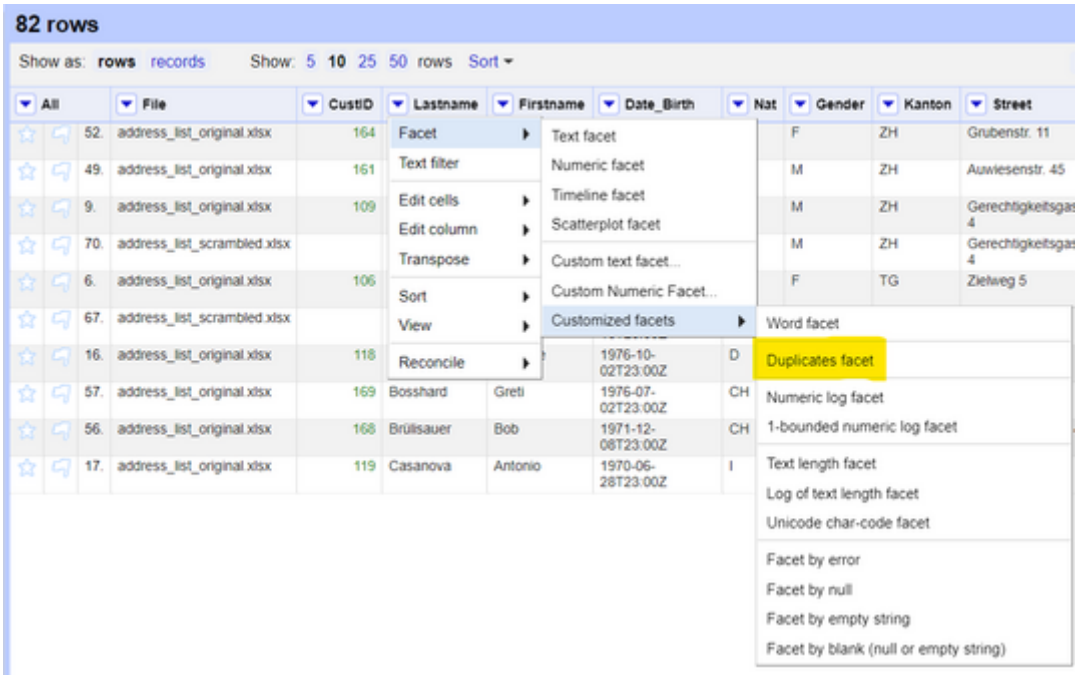


Abbildung 13. Option "Facetten duplizieren" in OpenRefine.

Diese Facette gibt `false` zurück, wenn der Wert kein Duplikat ist, und `true`, wenn der Wert ein Duplikat ist. In unserem Fall geht es darum, drei Facetten zu erstellen, die die Spalte `Firstname`, `Lastname` und `Date_Birth` auf Duplikate überprüfen.

Erstellen Sie drei Facetten für die drei Spalten und wählen Sie den Wert `true` für jede dieser Spalten. Dadurch werden die Spalten zurückgegeben, die nun in allen drei Werten der ausgewählten Spalten (unserem Ziel) doppelt vorhanden sind.

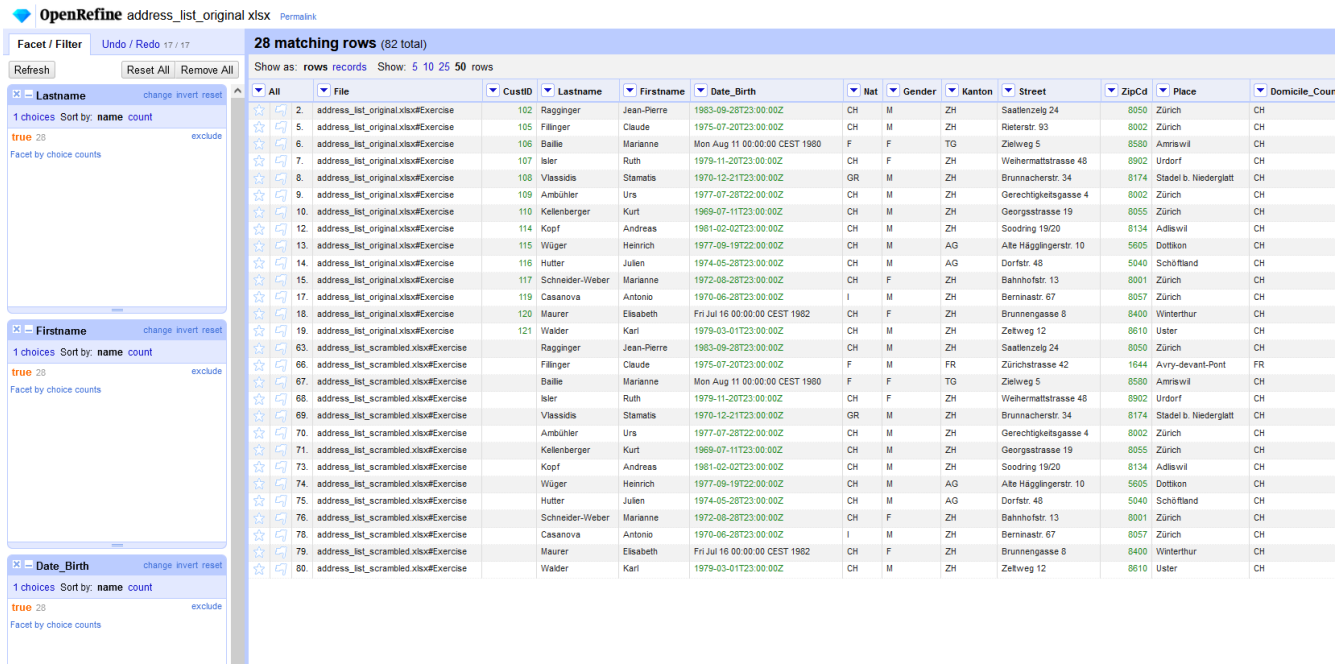


Abbildung 14. Anwenden aller doppelten Facetten auf den Datensatz.

Nun bleiben alle doppelten Datensätze übrig, d.h. gleicher Vorname, gleicher Nachname und gleiches Geburtsdatum. Die nächste Aufgabe besteht darin, die Datensätze alphabetisch nach einer der drei Spalten zu sortieren (z.B. nach `Firstname`). Nachdem Sie die Spalte sortiert haben, wählen Sie oben in der Menuleiste **[Sort]** und dann **[Reorder rows permanently]**. Dies ist notwendig,

um die nächste Funktion zu verwenden, die die doppelten Datensätze (ausser dem ersten) ausblendet.

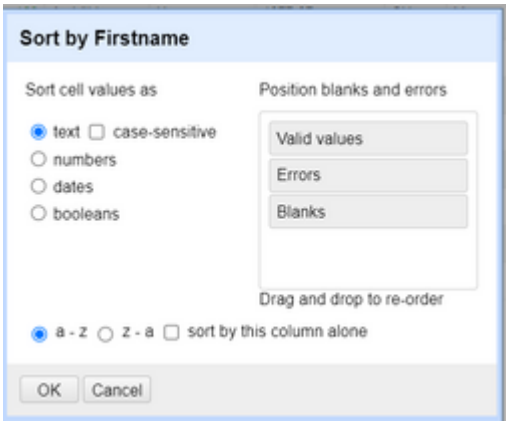


Abbildung 15. Sortierung nach der Spalte "Firstname".

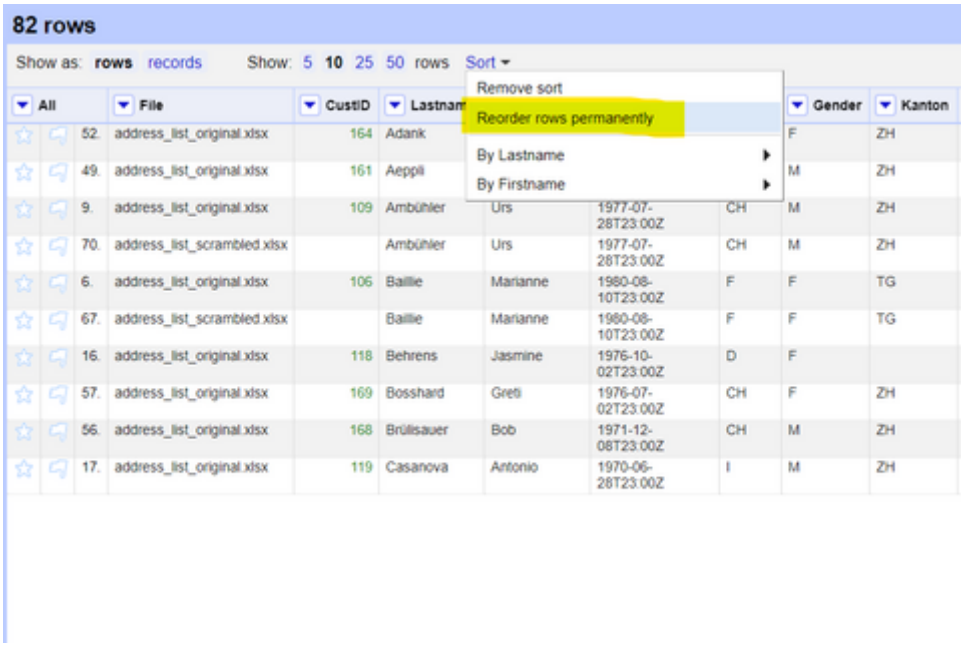


Abbildung 16. Persistieren der sortierten Werte.

Wenn Sie die Zeilen endgültig neu sortiert haben (ein notwendiger Schritt), markieren Sie die Spalte, die Sie sortiert haben, und klicken Sie auf **Edit Cells › Blank down**. Dadurch werden alle doppelten Datensätze (abgesehen von einem Eintrag) ausgeblendet, was genau das ist, was hier benötigt wird.

Jetzt müssen Sie die Zeilen nach Leerzeichen facettieren. Dies ist eine benutzerdefinierte OpenRefine-Facette, die Sie unter **Facet › Customized Facets › Facet by blank (null or empty string)** finden.

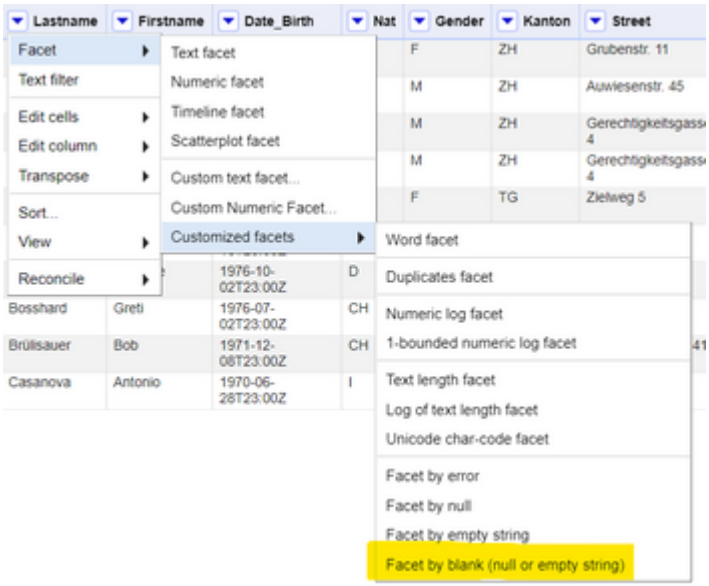


Abbildung 17. "Facet by blank" Option in OpenRefine.

Wählen Sie für die Facette, die mit der Option **[Facet by blank]** erstellt wurde, die Option `true` aus, damit nur noch die doppelten Datensätze (unser Ziel) übrig bleiben. Klicken Sie nun auf die Spalte *All* und wählen Sie **Edit Rows › Remove matching rows**. Dadurch werden alle Zeilen der aktuellen Auswahl (die doppelten Datensätze) entfernt.

Entfernen Sie alle Facetten, indem Sie auf **[Remove all]** klicken, und Sie erhalten nur noch eindeutige Datensätze.

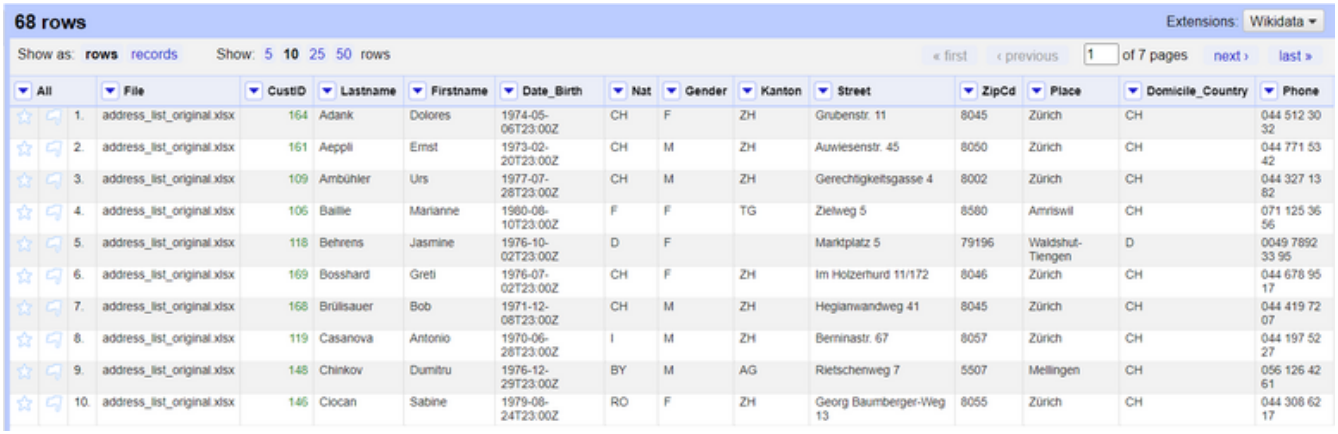


Abbildung 18. Datensätze nach Deduplizierung (und Validierung).

Schritt 3: Fertigstellung

In der Spalte `Phone` (oder `Phone_Number`) gibt es einige Telefonnummern mit Landesvorwahl und einige ohne. Es gibt auch Leerzeichen bei einigen der Nummern und bei anderen nicht.

Schreiben Sie eine Texttransformationsfunktion mit GREL für die `Phone`-Spalte, die alle Spaltenwerte standardisiert. Sie können dies tun, indem Sie entweder den Ländercode zu allen hinzufügen (die ihn nicht haben) oder indem Sie ihn von denen entfernen, die ihn haben. Sie müssen auch Leerzeichen aus den Werten entfernen, die sie haben, oder Leerzeichen zu den Werten hinzufügen, die sie nicht haben, das bleibt Ihnen überlassen.

Nachdem diese letzte Aufgabe erledigt ist, sind die Quelldaten erfolgreich in den Zieldatensatz

integriert, validiert und dedupliziert.

Schritt 4: Exportieren der Daten

Exportieren Sie die Daten in das Format *MS Excel 2007+ (.xslx)*, um diese Übung zu beenden.

5. Anreicherung von Daten mit Geokodierung

Geokodierung ist der Prozess der Konvertierung/Umwandlung einer vom Menschen lesbaren Beschreibung eines Ortes, z.B. einer Adresse oder eines Ortsnamens, in den tatsächlichen Standort des Ortes in der Welt (Geodaten). Die Geokodierung ist ein wichtiger Bestandteil von Geodaten und Standortanalysen. Die Idee der Geokodierung besteht darin, eine Beschreibung eines Ortes einzugeben und sich den genauen Standort ausgeben zu lassen (z.B. Längen- und Breitengrad, 'lat/lon'). **Umgekehrte Geokodierung** ist ein anderes (nicht so weit verbreitetes) Konzept, das das Gegenteil von Geokodierung ist, d.h. die Eingabe des genauen Standorts eines Ortes und die Ausgabe der Adresse oder des Ortes.

Die Geokodierung kann über Online-Webanwendungen oder Webdienste (APIs) erfolgen. Es gibt mehrere Geokodierungs-APIs, die Sie verwenden können und die in der Regel mit Kosten verbunden sind.

In diesem Arbeitsblatt wird demonstriert, wie die Geokodierung mit **Nominatim** (siehe Nominatim API, die auf der offenen Datenbank von OpenStreetMap basiert, funktioniert. Dies ist ein Beispiel für einen Nominatim-API-Aufruf für die Adresse "Obere Bahnhofstrasse 32b, Rapperswil" (aus der Dokumentation):
`"https://nominatim.openstreetmap.org/search?format=xml&addressdetails=1&countryCodes=CH&format=geojson&limit=1&q=32b+Obere+Bahnhofstrasse,+Rapperswil"`.

Dies ergibt das folgende GeoJSON (JSON)-Dokument (zu Schulungszwecken bearbeitet und gekürzt):


```
{
    "type": "FeatureCollection",
    "licence": "Data © OpenStreetMap contributors, ODbL 1.0.",
    "features": [{
        "type": "Feature",
        "properties": {
            "osm_id": 3124300001,
            "osm_type": "node",
            "importance": 0.42099999999999993
        },
        "geometry": {
            "type": "Point",
            "coordinates": [8.8190421, 47.2269746]
        }
    }]
}
```

Mehr zu JSON gibt es auf www.json.org.

Bitte beachten Sie die *Terms of Service* von kostenlosen Webservices wie der Nominatim API. Die Nominatim Usage Policy besagt zum Beispiel, dass maximal eine Anfrage pro Sekunde wiederholt werden darf. Das bedeutet, dass Sie Aufrufe mit einer Verzögerung von 1000ms drosseln müssen.

Aufgabe 3: Geocoding mit OpenRefine

In diesem Teil werden Sie Nominatim als Geokodierungsdienst zusammen mit OpenRefine verwenden. Da unser Datensatz Informationen über die Adresse und das Land des Kunden enthält, wird die Geocodierung an einem Datensatz Anwendung finden.

Diese Übung dient auch zur Veranschaulichung des zweiten Falls (*horizontal um eine oder mehrere Spalten erweitern*) des Kapitels *Datensätze aus verschiedenen Quellen verbinden*.

Daten

Für diese Aufgabe werden die Daten aus dem File `address_list_original.xlsx` verwendet.

Schritt 1: Erstellen des Projekts

Wenn Sie den richtigen Datensatz haben, öffnen Sie das Programm OpenRefine und erstellen Sie das Projekt mit dem gerade heruntergeladenen Datensatz.

Die Nominatim-API-Abfragen, die Breiten- und Längengrade ermitteln, können direkt in OpenRefine durchgeführt werden. Die Werte müssen dann nur noch in die jeweils richtige Spalte eingefügt werden.

Schritt 2: Verknüpfung der Adressspalten

Sie müssen die Adressspalten zu einer einzigen Spalte Full_Address zusammenführen, um für den Aufruf der Geokodierung bereit zu sein. Dies kann mithilfe von Spaltenverknüpfungen und anderen Funktionen wie folgt geschehen:

1. Erstellen Sie zunächst eine neue Spalte, die die Spalten `Street` und `Place` miteinander verknüpft. So erhalten Sie eine neue Spalte, die die Strasse und den Ort enthält.
2. Navigieren Sie zur Spalte `Street`, klicken Sie auf das Dropdown-Menü und wählen Sie **Edit Column › Join columns...**. Daraufhin öffnet sich ein Pop-up-Fenster, in dem Sie angeben können, welche Spalten Sie verbinden möchten und wie Sie sie verbinden wollen.
3. Wählen Sie die Spalten `Street` und `Address` auf der linken Seite aus und geben Sie als Trennzeichen `,` (ein Komma gefolgt von einem Leerschlag) zwischen den Inhalten der beiden Spalten auf der rechten Seite an. Aktivieren Sie ausserdem das Optionsfeld *Write result in new column named...* und geben Sie den Namen `Full_Address` an, da Sie aus diesen beiden Spalten eine neue Spalte erstellen wollen.

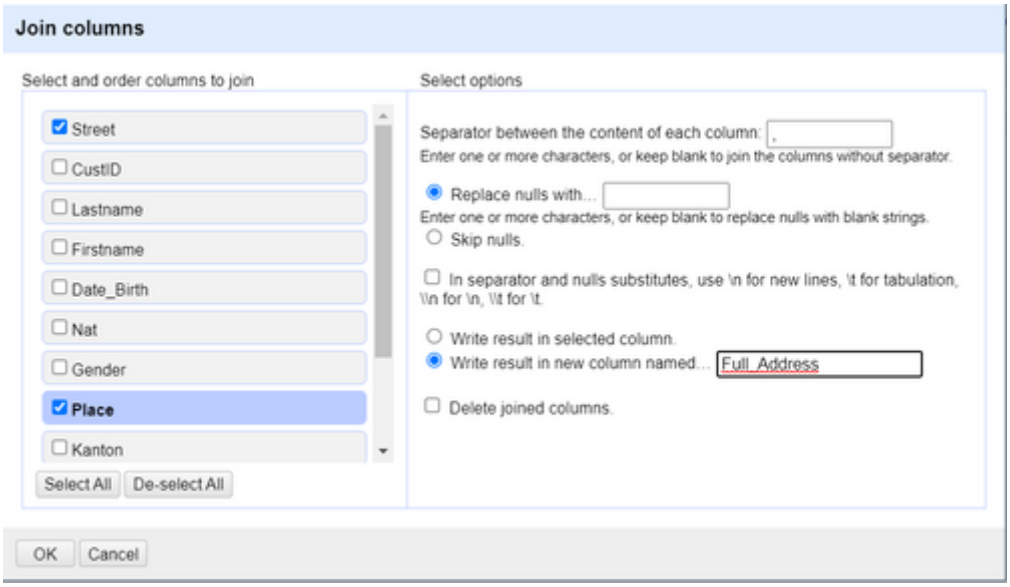
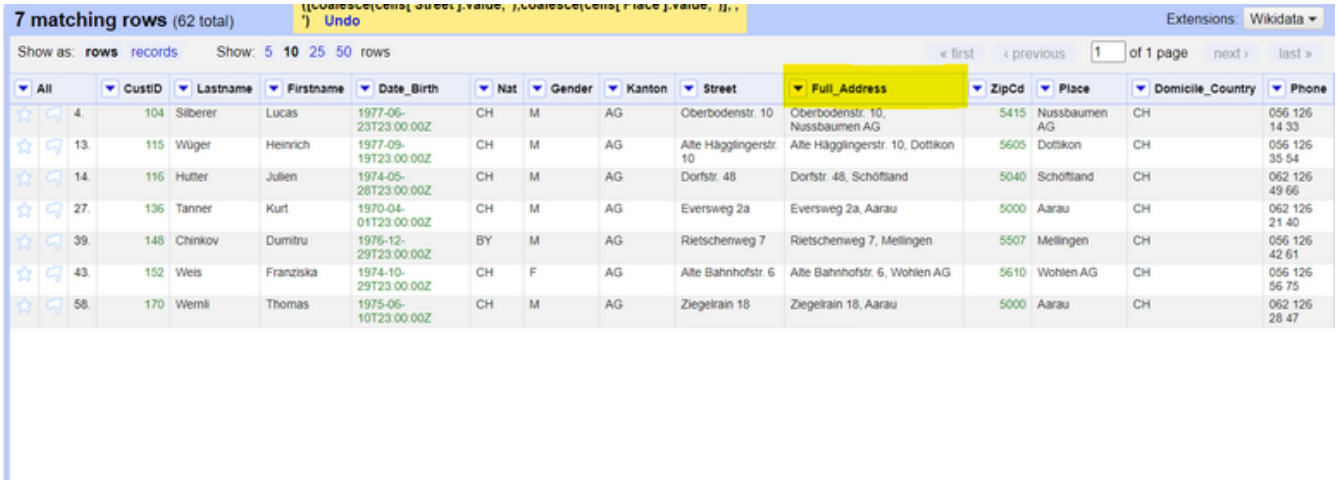


Abbildung 19. Verbindung/Verkettung von Spalten in OpenRefine.

4. Jetzt gibt es eine neue Spalte namens `Full_Address`, die die Adresse (mit Nummer) und den Ortsteil der Adresse enthält. Dies hilft der Geokodierungs-API bei der Bestimmung des Standorts der Kundenadressen.



All	CustID	Lastname	Firstname	Date_Birth	Nat	Gender	Kanton	Street	Full_Address	ZipCd	Place	Domicile_Country	Phone
4.	104	Silberer	Lucas	1977-06-23T23:00:00Z	CH	M	AG	Oberbodenstr. 10	Oberbodenstr. 10, Nussbaumen AG	5415	Nussbaumen AG	CH	056 126 14 33
13.	115	Wüger	Heinrich	1977-09-19T23:00:00Z	CH	M	AG	Alte Hägglingerstr. 10	Alte Hägglingerstr. 10, Dottikon	5605	Dottikon	CH	056 126 35 54
14.	116	Hutter	Julien	1974-05-28T23:00:00Z	CH	M	AG	Dorfstr. 48	Dorfstr. 48, Schöftland	5040	Schöftland	CH	062 126 49 66
27.	136	Tanner	Kurt	1970-04-01T23:00:00Z	CH	M	AG	Eversweg 2a	Eversweg 2a, Aarau	5000	Aarau	CH	062 126 21 40
39.	148	Chinkov	Dumitru	1976-12-29T23:00:00Z	BY	M	AG	Rietschenweg 7	Rietschenweg 7, Mellingen	5507	Mellingen	CH	056 126 42 61
43.	152	Weis	Franziska	1974-10-29T23:00:00Z	CH	F	AG	Alte Bahnhofstr. 6	Alte Bahnhofstr. 6, Wohlen AG	5610	Wohlen AG	CH	056 126 56 75
58.	170	Wemli	Thomas	1975-06-10T23:00:00Z	CH	M	AG	Ziegelrain 18	Ziegelrain 18, Aarau	5000	Aarau	CH	062 126 28 47

Abbildung 20. OpenRefine nach dem Verbinden der Spalten "Adress" und "Place".

Schritt 3: Verwendung einer Geocoding-API mit OpenRefine

Jetzt müssen Sie eine **Geokodierungs-API** mit OpenRefine aufrufen, die uns weitere Informationen über unsere Adresse, einschliesslich Lat/Lon-Attribute, liefert.

1. Sie müssen eine Anfrage an die Nominatim-API stellen, bei der Sie den Wert der Spalte *Full_Address* aufteilen und an die API senden, um im Gegenzug Informationen über diesen Ort zu erhalten, einschliesslich lat/lon (unser Ziel).
2. Klicken Sie zunächst auf die Dropdown-Schaltfläche der Spalte `Full_Address` und wählen Sie **Edit Column › Add column by fetching URLs...**.
3. Geben Sie einen neuen Namen für die Spalte an (`address_json` oder `osm_json`, es spielt keine Rolle), ändern Sie die *throttle delay* auf 1000ms und schreiben Sie den Ausdruck als: `'https://nominatim.openstreetmap.org/search?street=' + escape(value.split(",")[0], 'url') + '&city=' + escape(value.split(",")[1], 'url') + '&format=json'` Und klicken Sie auf **[OK]**.

Dieser Code teilt die Adressspalte in zwei Werte auf (einer enthält die Stadt, der andere den Ort) und stellt eine API-Anfrage, um ein **GeoJSON-Objekt** mit den Breiten- und Längenkoordinaten des Ortes zu erhalten (der Prozess der Geokodierung). Sie werden auch andere Informationen über den Ort erhalten, aber Sie können diese vorerst ignorieren.

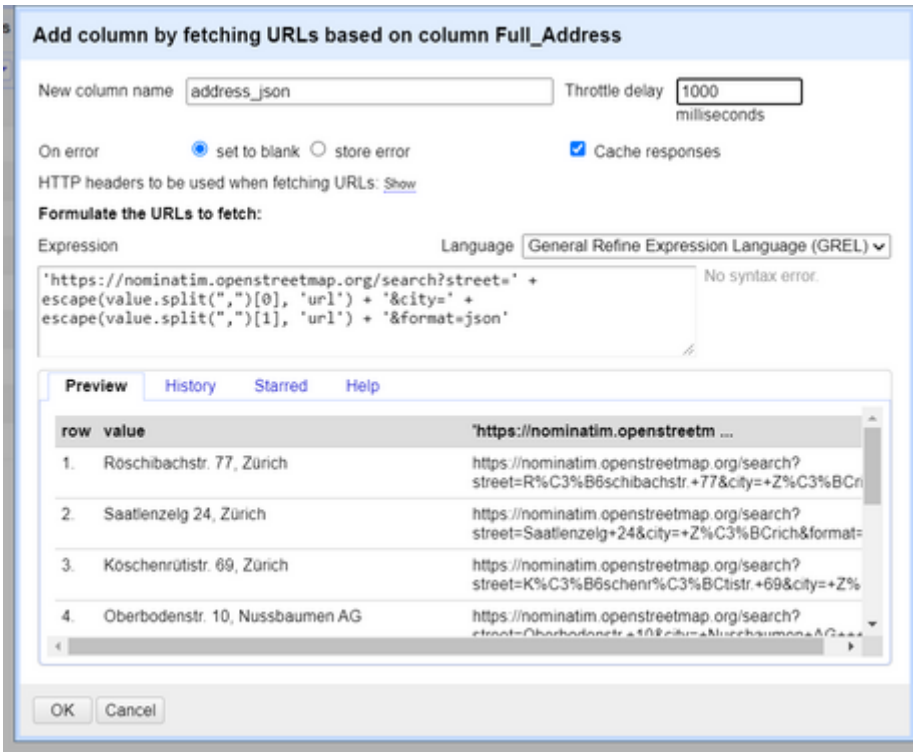


Abbildung 21. Hinzufügen einer Spalte durch Abrufen einer URL (API-Aufruf) mit GREL.

4. Jetzt haben Sie eine neue Spalte mit der API-Antwort für die Anfrage, die Sie gemacht haben. Wenn Sie diesen JSON-Code formatieren/verschönern, werden Sie sehen, dass er Informationen über den Standort enthält, einschliesslich Lat/Lon-Attribute, die Bounding Box, OpenStreetMap ID/Typ usw. wie oben gezeigt.

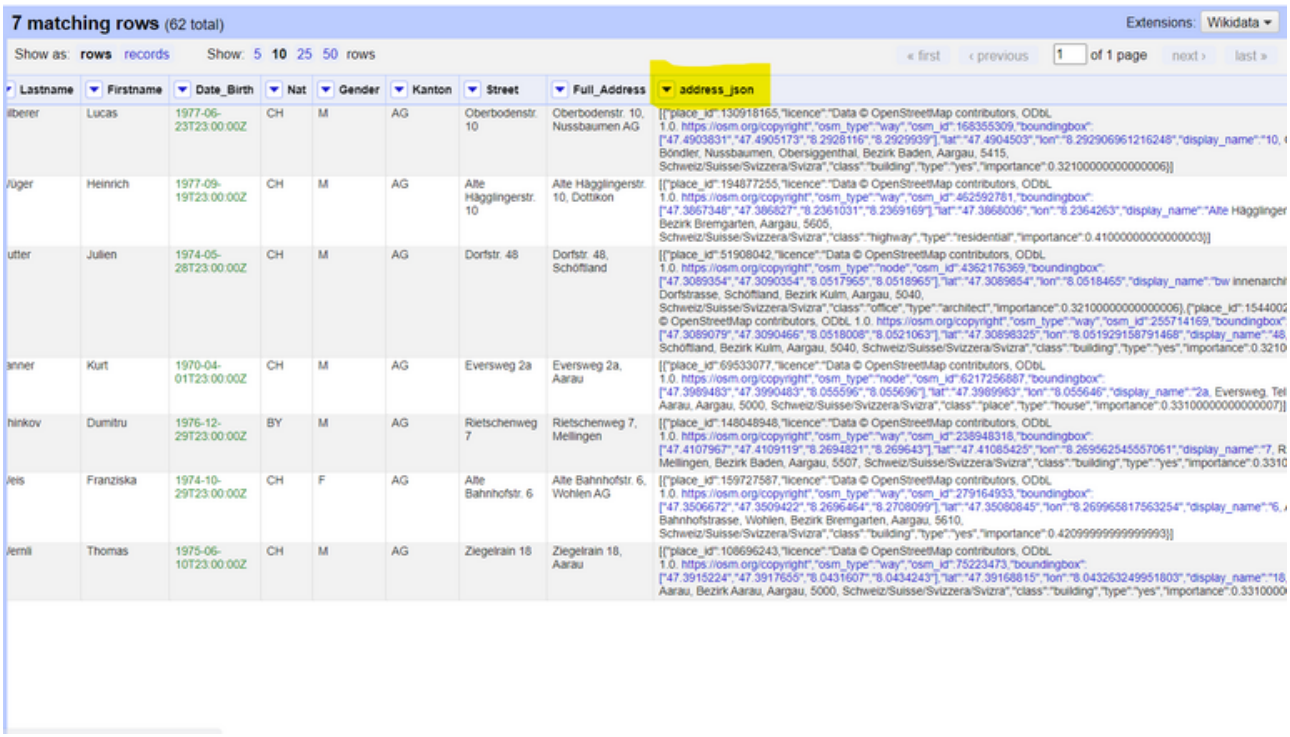


Abbildung 22. OpenRefine-Raster, nachdem die Spalte "address_json" hinzugefügt wurde.

Das Abrufen der Daten aus dem Internet kann einige Zeit in Anspruch nehmen, haben Sie also bitte etwas Geduld. Wenn der Vorgang schneller ablaufen soll, kann man den Datensatz mithilfe von Facetten auf weniger Datensätze eingrenzen, z.B. indem man nur die Datensätze mit dem Wert "AG" in der Spalte `Kanton` filtert.

Schritt 4: Parsen des JSON und Erstellen der Koordinatenspalten (lat/lon)

Nun, da Sie das JSON-Objekt mit Informationen über die Standorte der Kunden haben, können Sie dieses JSON-Objekt verwenden, um neue Spalten daraus zu erstellen, z.B. lat/lon.

Um das machen zu können werden folgende Schritte benötigt:

1. Klicken Sie auf die Dropdown-Liste der neuen Spalte, die Sie gerade durch Aufruf der API erstellt haben.
2. Klicken Sie auf **Edit column › Add column based on this column...**
3. Geben Sie einen Namen für die neue Spalte an (z.B. `lat` oder `latitude`) und verwenden Sie das Feld *Expression*, um das JSON zu parsen und das gewünschte Attribut zu lesen, indem Sie es eingeben: `value.parseJson()[0].lat`. Sie können die Vorschau der Spalte nach der Auswertung des Ausdrucks sehen. Klicken Sie auf **[OK]** und Sie sehen eine neue Spalte für die Breitengradkoordinate.

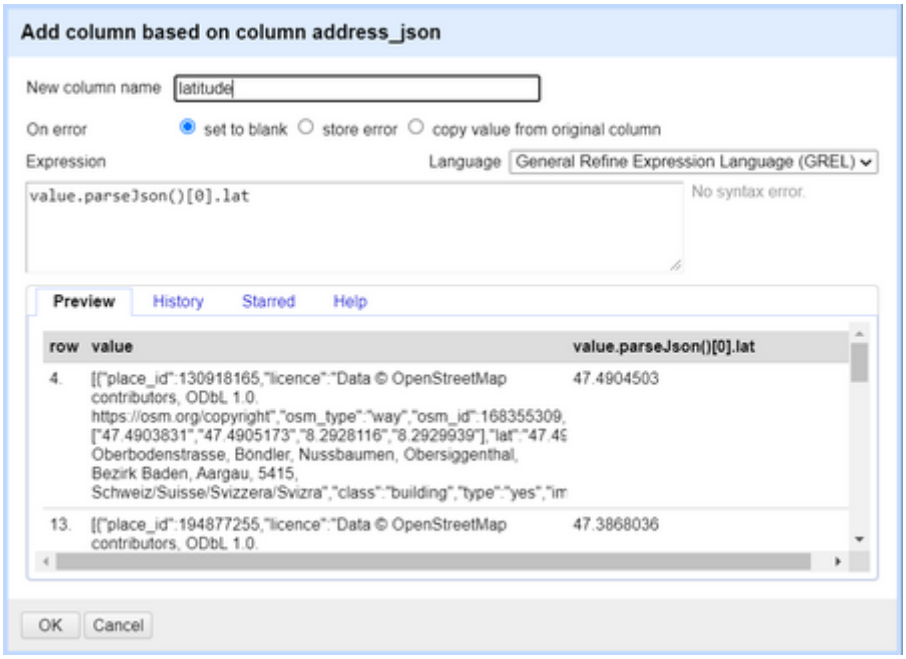
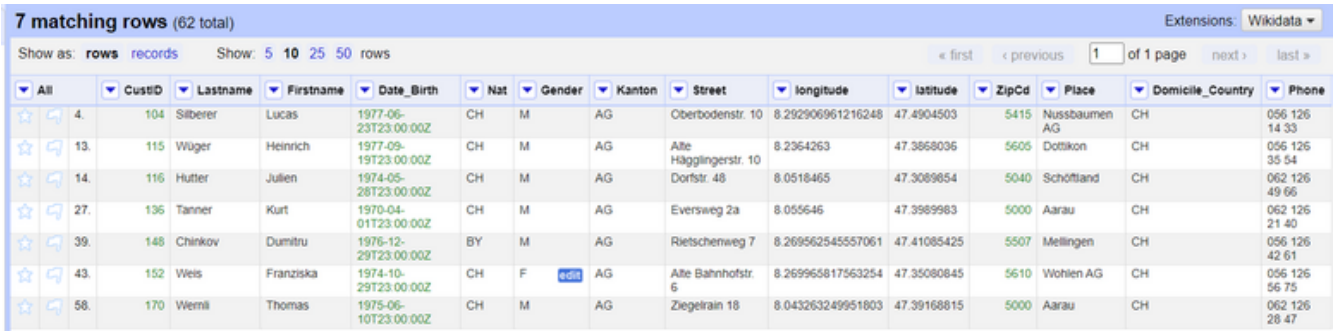


Abbildung 23. Extrahieren des Attributs "latitude" aus dem erhaltenen json-Objekt.

4. Das Gleiche gilt für die Längenkoordinate "lon".
5. Nachdem die lat/lon-Spalten vorhanden ist, kann die Spalte mit dem gesamten JSON-Objekt gelöscht werden, um einen besseren Überblick über alle Daten beizubehalten. Sie können auch die Spalte `Full_Address` entfernen, wenn Sie möchten, da sie nicht mehr erforderlich ist.

Jetzt haben Sie einen sauberen Datensatz mit *lat*- und *lon*-Koordinaten für die Kundenadressen, die mittels OpenRefine und Geokodierungs-APIs abgerufen wurden. Möglicherweise gibt es einige Orte, die von der API nicht gefunden wurden. In diesem Fall können Sie eine Facette verwenden, um die

leeren Werte herauszufiltern, und dann diese Adressen manuell ändern, um herauszufinden, was mit ihnen nicht stimmt, und dann die API-Anfrage erneut stellen.



7 matching rows (62 total) — Extensions: Wikidata

Show as: rows records — Show: 5 10 25 50 rows — « first ‹ previous 1 of 1 page next › last »

All	CustID	Lastname	Firstname	Date_Birth	Nat	Gender	Kanton	Street	longitude	latitude	ZipCd	Place	Domicile_Country	Phone
4.	104	Silberer	Lucas	1977-06-23T23:00:00Z	CH	M	AG	Oberbodenstr. 10	8.292906961216248	47.4904503	5415	Nussbaumen AG	CH	056 126 14 33
13.	115	Wüger	Heinrich	1977-09-19T23:00:00Z	CH	M	AG	Alte Hägglingerstr. 10	8.2364263	47.3868036	5605	Dottikon	CH	056 126 35 54
14.	116	Hutter	Julien	1974-05-28T23:00:00Z	CH	M	AG	Dorfstr. 48	8.0518465	47.3089854	5040	Schöftland	CH	062 126 49 66
27.	136	Tanner	Kurt	1970-04-01T23:00:00Z	CH	M	AG	Eversweg 2a	8.055646	47.3989983	5000	Aarau	CH	062 126 21 40
39.	148	Chinkov	Dumitru	1976-12-29T23:00:00Z	BY	M	AG	Rietschenweg 7	8.269562545557061	47.41085425	5507	Mellingen	CH	056 126 42 61
43.	152	Weis	Franziska	1974-10-29T23:00:00Z	CH	F edit	AG	Alte Bahnhofstr. 6	8.269965817563254	47.35080845	5610	Wohlen AG	CH	056 126 56 75
58.	170	Wernli	Thomas	1975-06-10T23:00:00Z	CH	M	AG	Ziegelrain 18	8.043263249951803	47.39168815	5000	Aarau	CH	062 126 28 47

Abbildung 24. Endgültiges Aussehen des Datensatzes nach der Geokodierung und Bereinigung.

6. Web scraping

In diesem Abschnitt werden Sie Web Scraping mit OpenRefine durchführen. Mithilfe von Funktionen können Sie zunächst eine Webseite, die aus HTML-Text besteht, abrufen und parsen und dann die Daten nach den gewünschten Spalten filtern.

Die **GREL-Funktionen** von OpenRefine ermöglichen es uns, die HTML-Seite in HTML-Textinhalte zu zerlegen und dann verschiedene Methoden zur Auswahl der richtigen Tags, Attribute, Textknoten usw. zu verwenden.

Lassen Sie uns zuerst eine typische HTML-Struktur erklären und dann, was **GREL** ist.

HTML und DOM

HTML ist die **Standard-Markup-Language** für Webseiten. Es handelt sich nicht um eine Programmiersprache, sondern um eine interpretierte Sprache, die die Struktur einer Webseite beschreibt.

Webseiten werden in der HyperText Markup Language (HTML) geschrieben. HTML ist die Standardsprache für Dokumente, die in einem Webbrowser angezeigt werden sollen.

Das **Document Object Model (DOM)** ist die Datendarstellung der Elemente/Objekte, die zum Inhalt eines HTML-Dokuments beitragen. Das DOM spielt beim Web Scraping eine entscheidende Rolle, da es für den Zugriff auf Elemente innerhalb einer Webseite verwendet werden kann (genau wie die DOM-Selektormethoden in JavaScript).

Mehr über HTML finden Sie hier und über Javascript DOM hier.

General Refine Expression Language (GREL)

Ein Hauptmerkmal von OpenRefine ist die **General Refine Expression Language**, kurz **GREL**. GREL ist eine OpenRefine-spezifische Ausdruckssprache, die ähnlich wie JavaScript funktioniert und für die Ausführung verschiedener Funktionen geeignet ist, wie z.B.:

- String-Operationen

- Boolean Operatoren
- Parsen von HTML, JSON oder XML
- Auswahl von HTML-Elementen
- Iteration über Elemente usw.

GREL wird in unserem Fall verwendet, um auf das **HTML-DOM** zuzugreifen und die entsprechenden Daten aus der HTML-Seite für das Scraping auszuwählen.

Mehr Informationen zu GREL und seinen Funktionen gibt es auf der Dokumentation von OpenRefine: GREL und GREL functions.

Aufgabe 4: Web scraping mit OpenRefine

Diese Übung beschäftigt sich mit:

- Den HTML-Inhalt in OpenRefine bringen.
- HTML und seine Elemente mit der OpenRefine-Ausdruckssprache GREL parsen.
- Anordnen der resultierenden Spalten.

Sie werden eine Wikipedia-Seite scrapen, die eine Liste aller Burgen im Kanton *Aargau* in der Schweiz enthält, wie zum Beispiel die *Habsburg*! Zuerst werden Sie den reinen **HTML-Inhalt** dieser Seite in OpenRefine einbringen und dann verschiedene **GREL HTML- und Text-Transformationsfunktionen** verwenden, um die richtigen Daten aus dem HTML-Inhalt herauszulesen.

Dies ist die Wikipedia-Webseite, die Sie für diese Übung verwenden werden: "Liste von Burgen und Schlössern im Kanton Aargau".

Schritt 1: Erstellen des "HTML"-Projekts

Es gibt mehrere Möglichkeiten, den HTML-Inhalt einer Webseite in OpenRefine einzubinden. Für diese Übung können Sie jedoch die Option **[Clipboard]** von OpenRefine verwenden. Kopieren Sie den Link der Webseite und fügen Sie ihn in den Textbereich der *Zwischenablage* auf der Startseite von OpenRefine ein.

Verwenden Sie den folgenden Link `https://de.wikipedia.org/wiki/Liste_von_Burgen_und_Schl%C3%B6ssern_im_Kanton_Aargau`. Openrefine kann nicht in allen Fällen URLs mit Umlauten verarbeiten, weshalb in diesem Link die Umlaute escaped wurden.

Abbildung 25. Verwendung der "Clipboard"-Option zum Erstellen eines Projekts in OpenRefine.

Konfiguration der Parsing-Optionen: (auf "standard" belassen), Projekt bennenen und auf **[Create Project]** klicken.

Schritt 2: Einbringen des HTML-Inhalts in OpenRefine

Bis zu diesem Punkt existiert nur eine Spalte mit einer Zelle, in welcher sich der Link befindet.

Abbildung 26. Die einzelne Spalte mit der URL der Webseite, die gescraped werden soll.

Klicken Sie nun auf das Dropdown-Menü dieser Spalte und wählen Sie **Edit Column › Add column by fetching URLs...**. Geben Sie der neuen Spalte einfach einen Namen, z.B. `html` oder `html_content`, lassen Sie den Ausdruck als `value` stehen und klicken Sie auf **[OK]**.

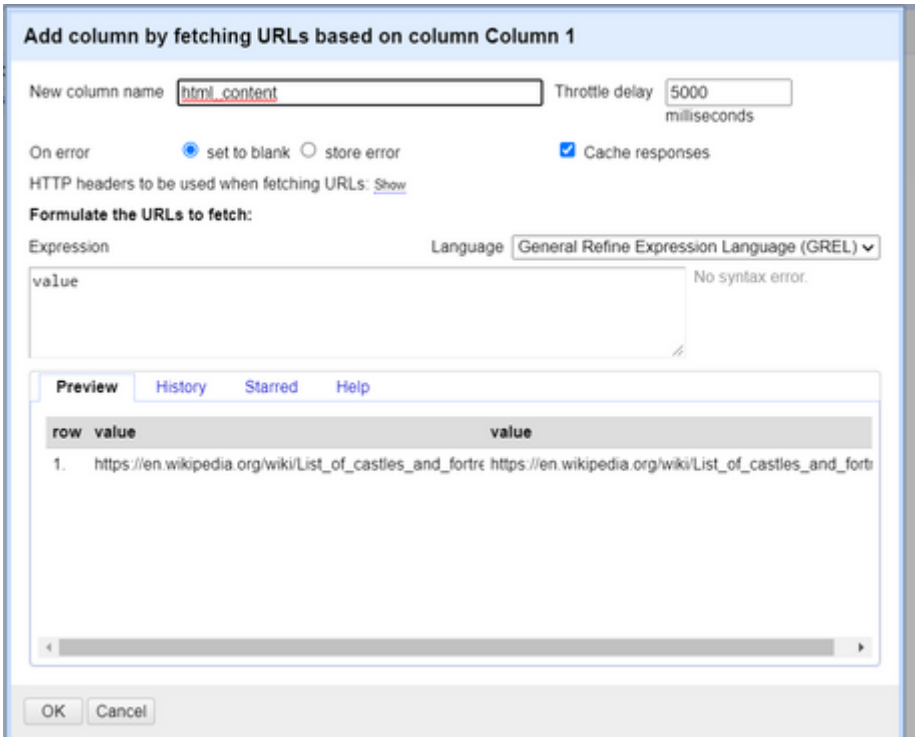


Abbildung 27. Hinzufügen des gesamten HTML-Inhalts in eine neue Spalte mithilfe der OpenRefine-Option "Add column by fetching URL".

Nun wird eine neue Spalte erstellt, die den gesamten reinen HTML-Inhalt der ausgewählten Webseite enthält. Sie sollten diese Spalte verwenden, um die notwendigen Informationen zu extrahieren, die Sie für die Daten der Schlösser benötigen.

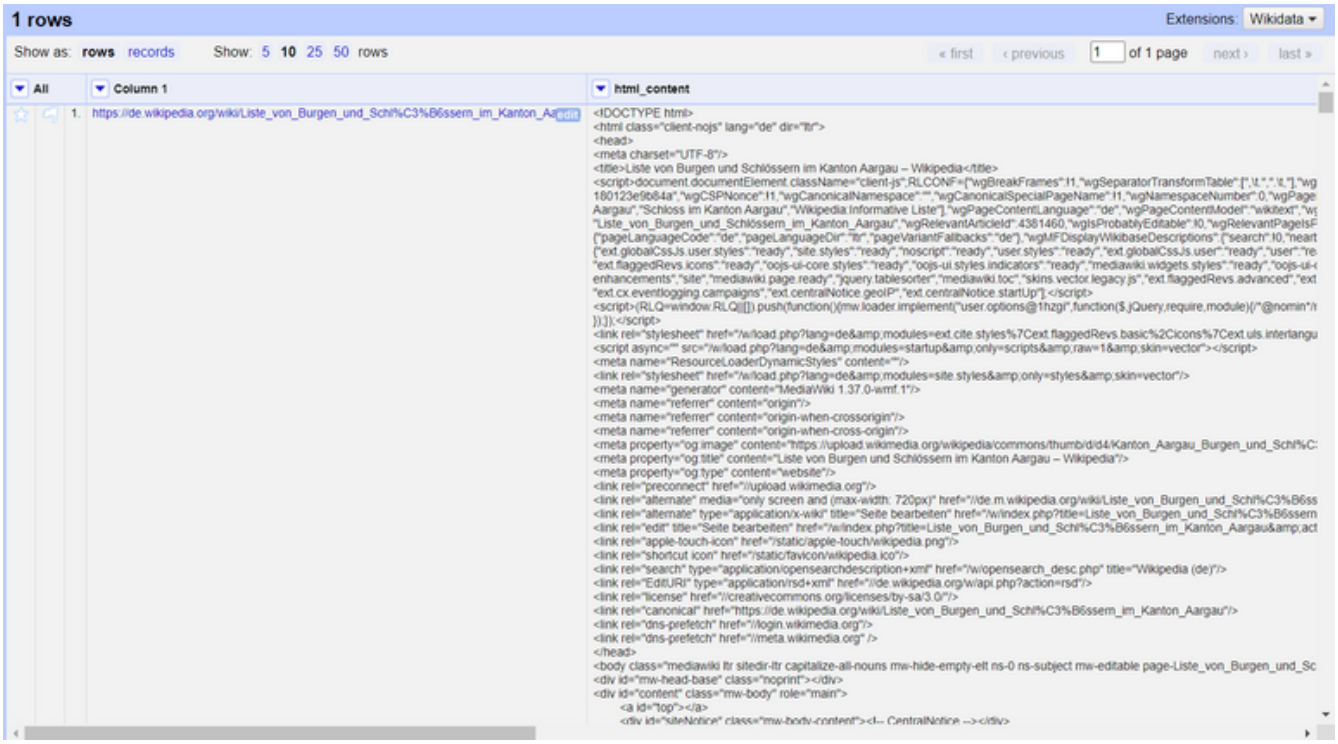


Abbildung 28. Die Spalte "html_content", die den vollständigen HTML-Inhalt enthält, nachdem die Spalte durch Abrufen der URL hinzugefügt wurde.

Schritt 3: Parsen des HTML und seiner Elemente mit GREL

Nun werden die HTML-Parsing-Funktionen von GREL verwendet, um den HTML-Code zu

analysieren und in die entsprechenden Spalten einzuordnen.

Hier bietet es sich an eine `forEach` iterative Schleife von GREL zu verwenden, um die entsprechenden Elemente zu durchlaufen und nur die Daten über die Schlösser zu extrahieren. In der Schleife werden die HTML-Parsing-Funktionen von GREL und grundlegende CSS-Selektoren-Logik verwendet, um die benötigten HTML-Elemente auszuwählen.

1. Klicken Sie auf das Dropdown-Menü der soeben erstellten HTML-Inhaltsspalte und klicken Sie auf **Edit column › Add column based on this column...**.
2. Geben Sie den folgenden Code in das Ausdrucksfeld ein:

```
forEach(value.parseHtml().select("table.wikitable tbody tr td:first-child > a"), e,
e.ownText()).join("|")
```

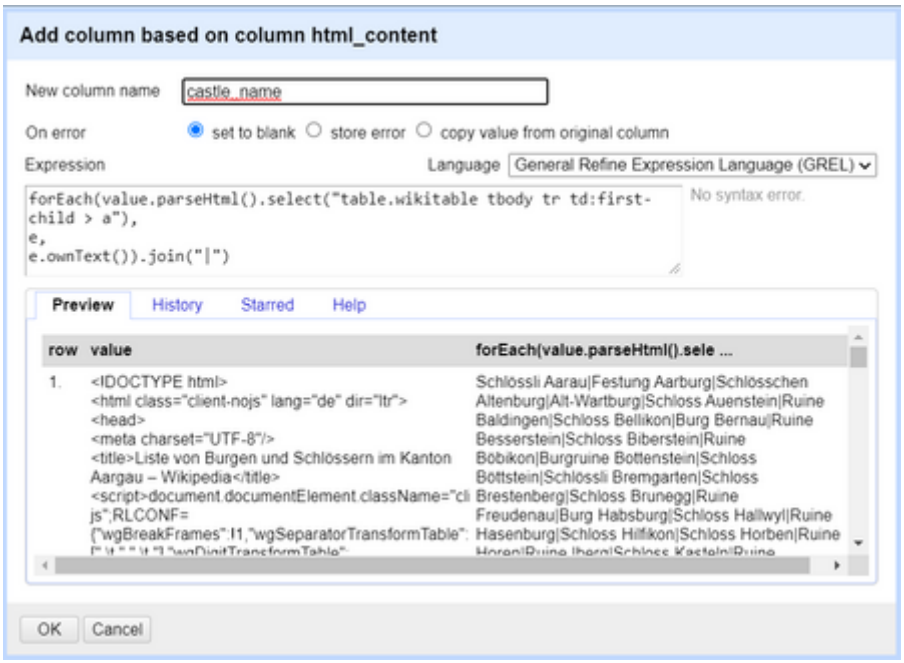


Abbildung 29. Verwendung von GREL zum Herausfiltern der benötigten Daten aus dem HTML-Inhalt.

3. Dieser Code durchläuft die Elemente, die im ersten Argument der `forEach`-Schleife angegeben sind, unter Verwendung grundlegender CSS-Selektoren, gibt jedem Element in der aktuellen Iteration der Schleife im zweiten Argument einen Namen (in unserem Fall: `e`) und führt dann für jedes Element in der Schleife eine Aktion aus (`e.ownText()`). Nach Beendigung der Schleife erhalten Sie ein Array dieser Werte, das Sie mit einem `|`-Trennzeichen verbinden und dann in mehrere Zellen aufteilen.
4. Es bleibt eine neue Spalte übrig, die eine Zeichenkette mit allen Namen der Schlösser enthält, die durch ein "|" Zeichen verbunden sind.
5. Jetzt müssen Sie die erhaltene Zeichenkette in mehrere Zellen aufteilen, indem Sie sie mit dem Trennzeichen `|` trennen (der Grund, warum dieser Schritt überhaupt gefordert war). Klicken Sie auf das Dropdown-Menü der soeben erstellten Spalte und klicken Sie auf **Edit Cells › Split multi-valued cells...**. Wählen Sie das Trennzeichen `|` und klicken Sie auf **[OK]**.

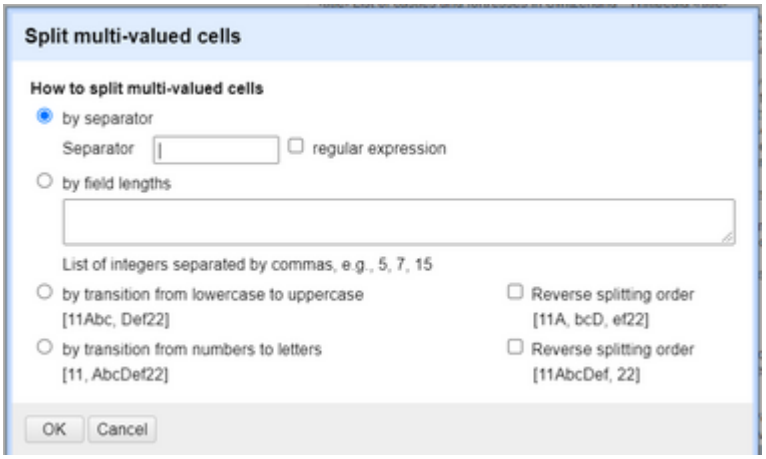


Abbildung 30. Verwenden Sie die Option "Split multi-valued cells", um die zuvor mit GREL erstellte Zeichenfolge aufzuteilen.

6. Ihre neue Spalte sollte nun für jede Burg eine Zeile enthalten (mit dem Namen der Burg).
7. Fahren Sie nun mit dem Extrahieren der anderen notwendigen Informationen fort, wie z.B.: location, type, date, notes, usw. Die Logik ist dieselbe wie oben, aber Sie müssen möglicherweise die CSS-Selektoren und die mit der Schleifenvariablen durchgeführte Aktion ändern (z.B. `e.htmlAttr("title")` extrahiert das Titelattribut des Links usw.).

Wenn Sie alle notwendigen Informationen zu den Schlössern extrahiert haben, bereinigen Sie Ihren Datensatz, indem Sie die HTML-Inhaltsspalten und andere unnötige Daten/Spalten entfernen. Übrig bleibt eine Tabelle mit den Schlössern und Festungen der Schweiz, die Sie mit OpenRefine aus Wikipedia extrahiert haben.

Es gibt mehrere Möglichkeiten, HTML-Inhalte zu extrahieren, aber sobald Sie alle Informationen, die Sie benötigen, effektiv erhalten können, sind sie alle gültig.

49 rows

Show as: **rows** records Show: 5 10 **25** 50 rows

All		name	ortschaft	jahr	typ	zustand
	1.	Schlössli Aarau	Aarau	1240	Schloss	erhalten
	2.	Festung Aarburg	Aarburg	1123	Festung	erhalten
	3.	Schlösschen Altenburg	Altenburg bei Brugg	370	Kastell	erhalten
	4.	Alt-Wartburg	Oftringen	1200	Burg	Ruine
	5.	Schloss Auenstein	Auenstein AG	1307	Schloss	erhalten
	6.	Ruine Baldingen	Baldingen AG	unbekannt	Burg	verfallen
	7.	Schloss Bellikon	Bellikon	1430-1440	Schloss	erhalten
	8.	Burg Bernau	Leibstadt	1157	Burg	Ruine
	9.	Ruine Besserstein	Villigen	unbekannt	Burg	verfallen
	10.	Schloss Biberstein	Biberstein	1280	Schloss	erhalten
	11.	Ruine Böbikon	Böbikon	unbekannt	Burg	Ruine
	12.	Burgruine Bottenstein	Zofingen	Mitte 13. Jahrhundert	Burg	Ruine
	13.	Schloss Böttstein	Böttstein	1100-1200	Schloss	erhalten
	14.	Schlössli Bremgarten	Bremgarten AG	1238	Schloss	erhalten
	15.	Schloss Brestenberg	Seengen	1625	Schloss	erhalten
	16.	Schloss Brunegg	Brunegg	1250	Schloss	erhalten
	17.	Ruine Freudenau	Untersiggenthal	1240	Burg	Ruine
	18.	Burg Habsburg	Habsburg AG	1030	Burg	erhalten
	19.	Schloss Hallwyl	Seengen	1265	Schloss	erhalten
	20.	Ruine Hasenburg	Bergdietikon	unbekannt	Burg	verfallen
	21.	Schloss Hilfikon	Hilfikon	unbekannt	Schloss	erhalten
	22.	Schloss Horben	Beinwil (Freiamt)	1700	Schloss	erhalten
	23.	Ruine Horen	Küttigen	unbekannt	Burg	Ruine
	24.	Ruine Iberg	Riniken	11. Jahrhundert	Burg	verfallen
	25.	Schloss Kasteln	Oberflachs	1238	Schloss	erhalten

Abbildung 31. Datensatz der Schweizer Schlösser nach dem Scraping der Webseite, der Umwandlung und der Bereinigung der Daten in OpenRefine.

Es gibt einige GREL-Funktionen, die notwendig sind, um einige der HTML-Informationen aus dem HTML-Inhalt hier zu extrahieren. Sie können sich diese im obigen Abschnitt über GREL ansehen.

7. Schlussfolgerung und Ausblick

Anhand der oben besprochenen und erläuterten Punkte können Sie sehen, dass OpenRefine ein praktisches Werkzeug für die Arbeit mit Daten ist, seien diese nun unordentlich oder nicht. Allerdings ist es wichtig, die richtigen Funktionen zu kennen, wenn man mit Daten arbeitet!

Für die Arbeit mit Daten und die Umwandlung von Daten in etwas Sinnvolles und Nützliches ist eine Menge logisches, technisches und praktisches Wissen erforderlich. In den Übungen haben Sie hoffentlich gelernt, wie Sie mit unübersichtlichen Daten umgehen und welche Funktionen/Werkzeuge Sie bei verschiedenen Datenproblemen wählen sollten.

8. Was gelernt wurde

- Erstellen eines Projekts in OpenRefine.
- Erkunden eines Datensatzes und dessen Transformation mit OpenRefine, unter Verwendung von Facetten und anderen Transformationsfunktionen.
- Verwendung von OpenRefine zur Bereinigung/Duplizierung und Integration eines Datensatzes in einen anderen.
- Verwendung von OpenRefine zur Geokodierung und zum Web Scraping.

Falls Sie an der Integration von Geodaten wie beispielsweise Points of Interest interessiert sind, dann empfehlen wir, die OpenRefine-Extensions ‘OSM Extractor’ und ‘GeoJSON Export’ zu installieren. Diese findet man auf der Seite OpenRefine Downloads.

Was Sie hier nicht gelernt haben, sind viele weitere eingebaute Funktionen von OpenRefine (z.B. für Datums-Angaben) sowie komplexere Analysen, wie statistische Datenbeschreibungen (Mittelwert, Median), und vor allem, was Datenabgleich mit OpenRefine ist. Um mehr darüber zu erfahren, schauen Sie sich folgende Ressourcen an.

Empfohlene Lektüre

Das Buch "Using OpenRefine" von Verborgh & De Wilde (Packt Publishing Ltd, 2013) ist online frei verfügbar. Ausserdem gibt es viele Online-Tutorials und Videos, wie zum Beispiel die Library Carpentry OpenRefine lesson oder das Introduction video by the University of Idaho Library.

9. APPENDIX: Übungsfragen

Dies sind Übungen und Fragen zum gelernten Stoff und eine Art Wiederholungsfragen zu OpenRefine und Datenintegration.

Frage 1: Sammeln und notieren Sie alle wichtigen OpenRefine-Funktionen, die in ÜBUNG 1 genannt wurden.

Frage 2: Sammeln und notieren Sie alle wichtigen OpenRefine-Funktionen, die in ÜBUNG 2 genannt wurden.

Frage 3: Was sind die Herausforderungen der Datenintegration? Fassen Sie diese in drei Sätzen zusammen und geben Sie jeweils ein Beispiel.

Frage 4: Nennen Sie einige Transformationsfunktionen von OpenRefine.

Frage 5: Welche OpenRefine-Funktion oder welcher OpenRefine-Prozess ist äquivalent zu einem SQL-Join?

Frage 6: Welcher OpenRefine-Funktion entspricht der SQL-Befehl `unnest` am ehesten?

Noch Fragen? Sehen Sie auch "Kontakt" auf OpenSchoolMaps!