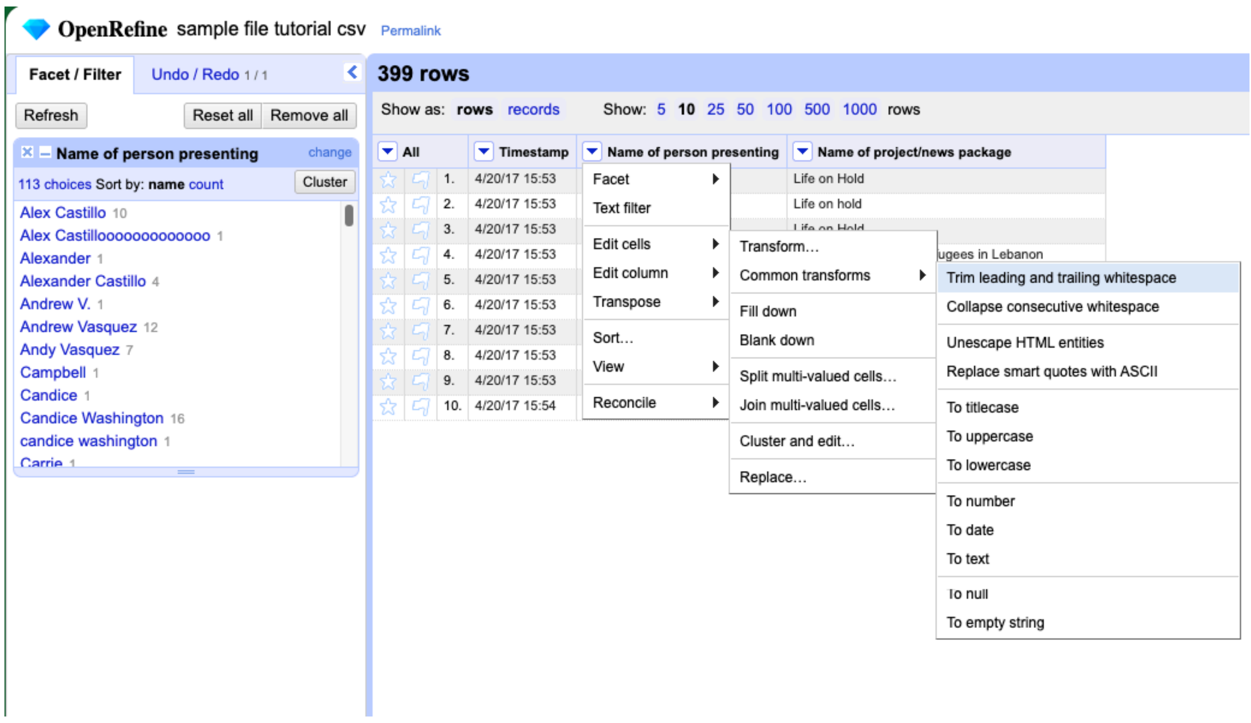
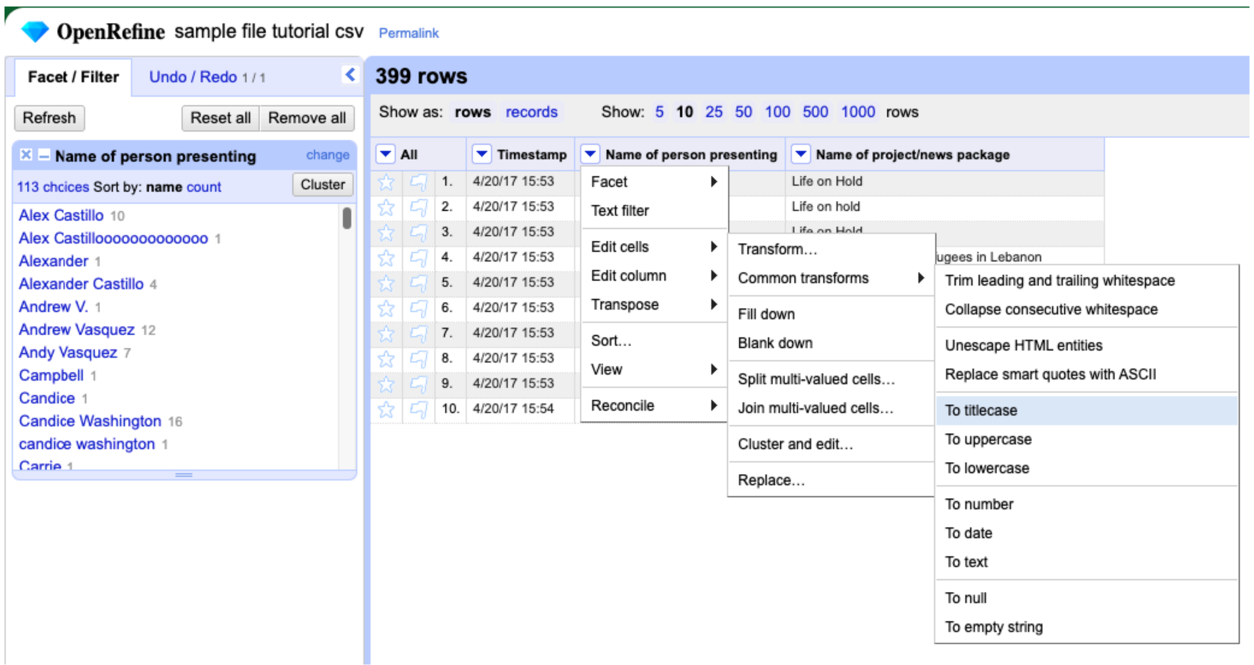
Business Intelligence - E07 - ETL (Open Refine)

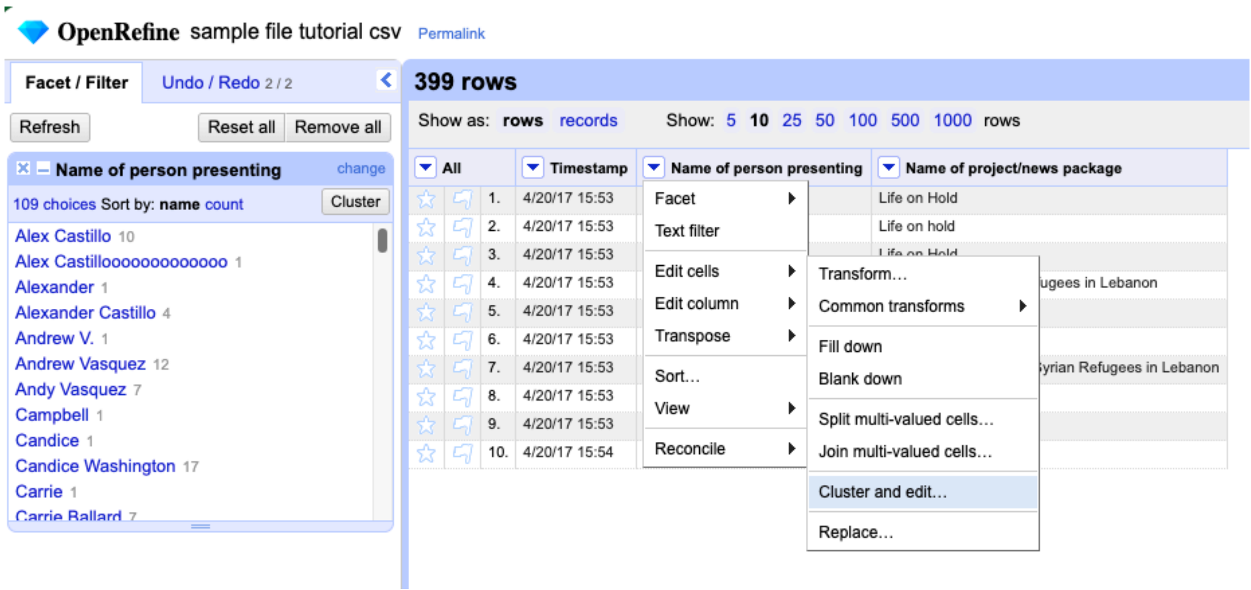
Ömer Furkan Çoban

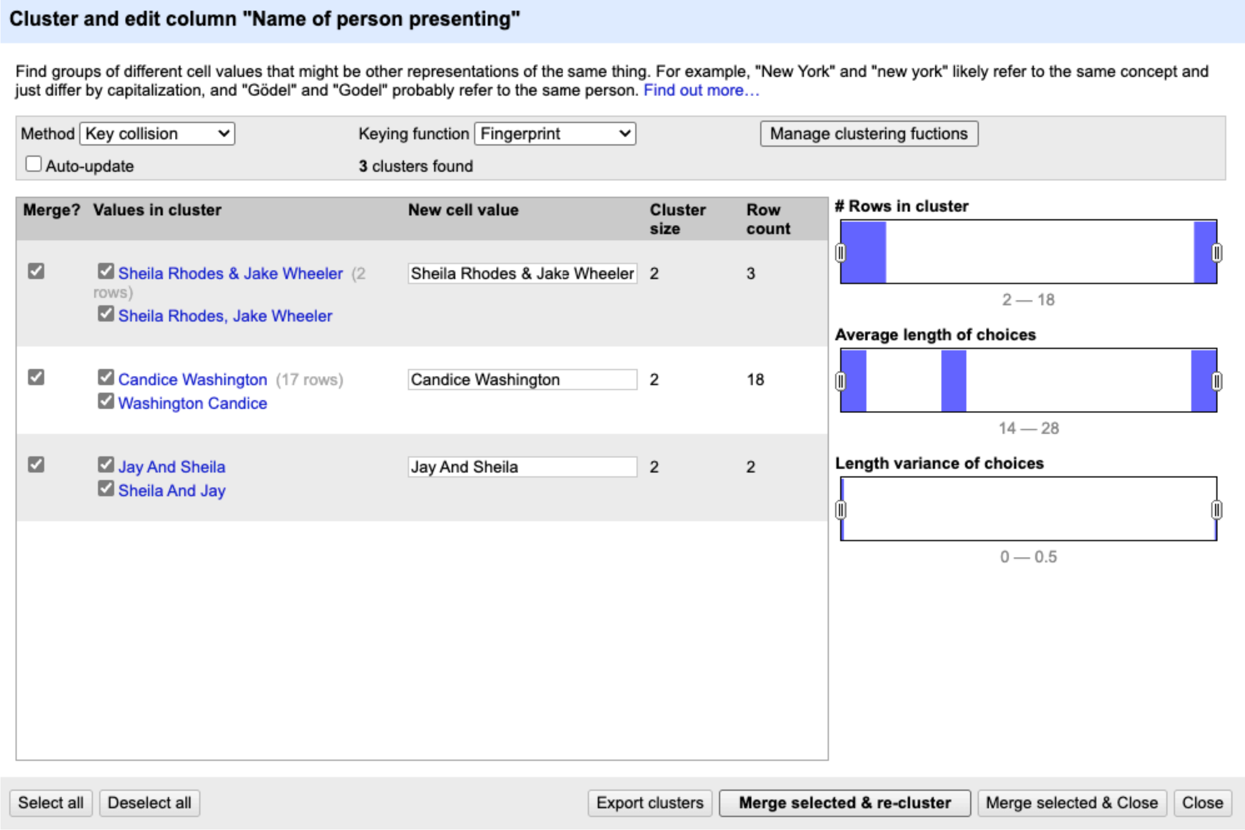
Tasks - Part I

1.In order to get familiar with the tool follow this tutorial on OpenRefine“Using OpenRefine to Clean Your Data”–https://multimedia.report/tutorials/openrefine/









Cluster and edit column "Name of person presenting"

Find groups of different cell values that might be other representations of the same thing. For example, "New York" and "new york" likely refer to the same concept and just differ by capitalization, and "Gödel" and "Godel" probably refer to the same person. Find out more…

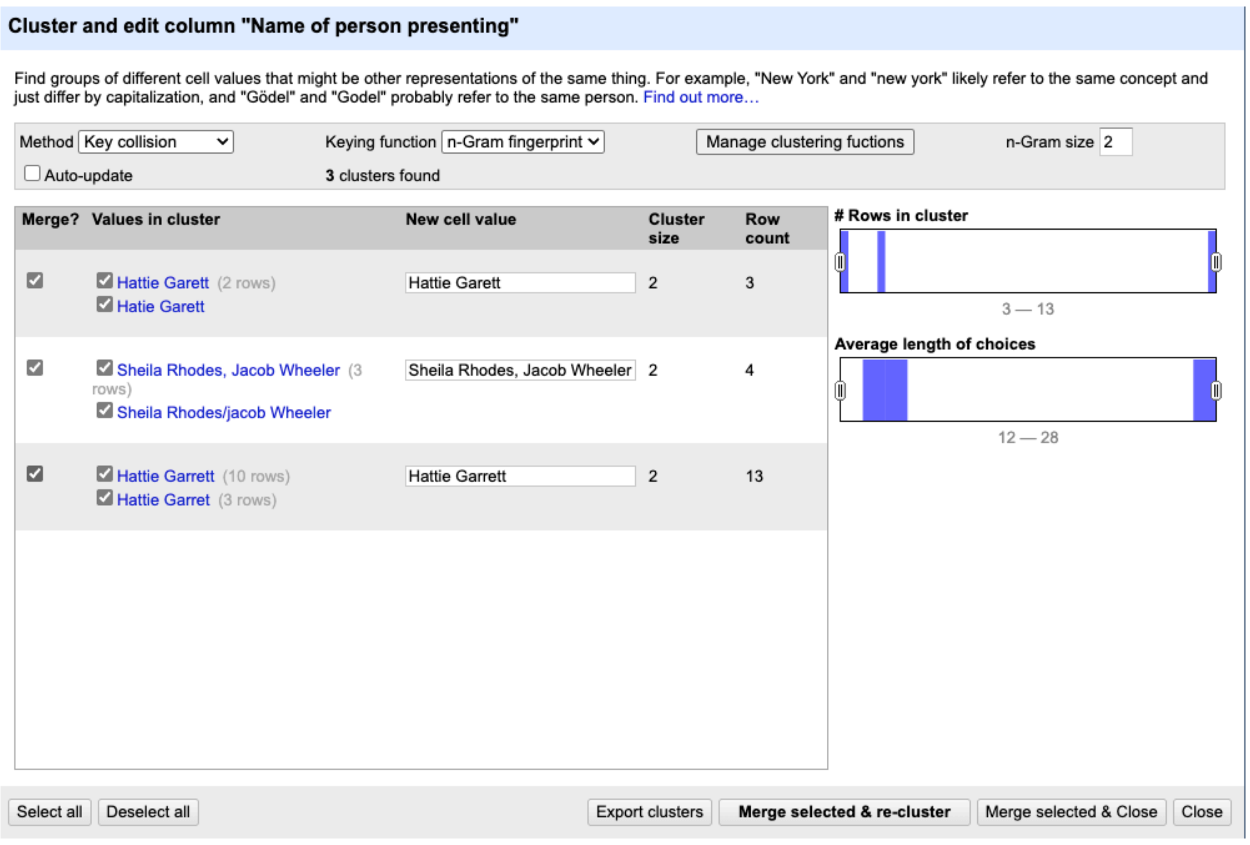
Method: Key collision | Keying function: Fingerprint

□ Auto-update

3 clusters found

Merge?	Values in cluster	New cell value	Cluster size	Row count
☑	☑ Sheila Rhodes & Jake Wheeler (2 rows) ☑ Sheila Rhodes, Jake Wheeler	Sheila Rhodes & Jake Wheeler	2	3
☑	☑ Candice Washington (17 rows) ☑ Washington Candice	Candice Washington	2	18
☑	☑ Jay And Sheila ☑ Sheila And Jay	Jay And Sheila	2	2

Manage clustering fuctions
# Rows in cluster
2 — 18
Average length of choices
14 — 28
Length variance of choices
0 — 0.5
Select all
Deselect all
Export clusters
Merge selected & re-cluster
Merge selected & Close
Close



Cluster and edit column "Name of person presenting"

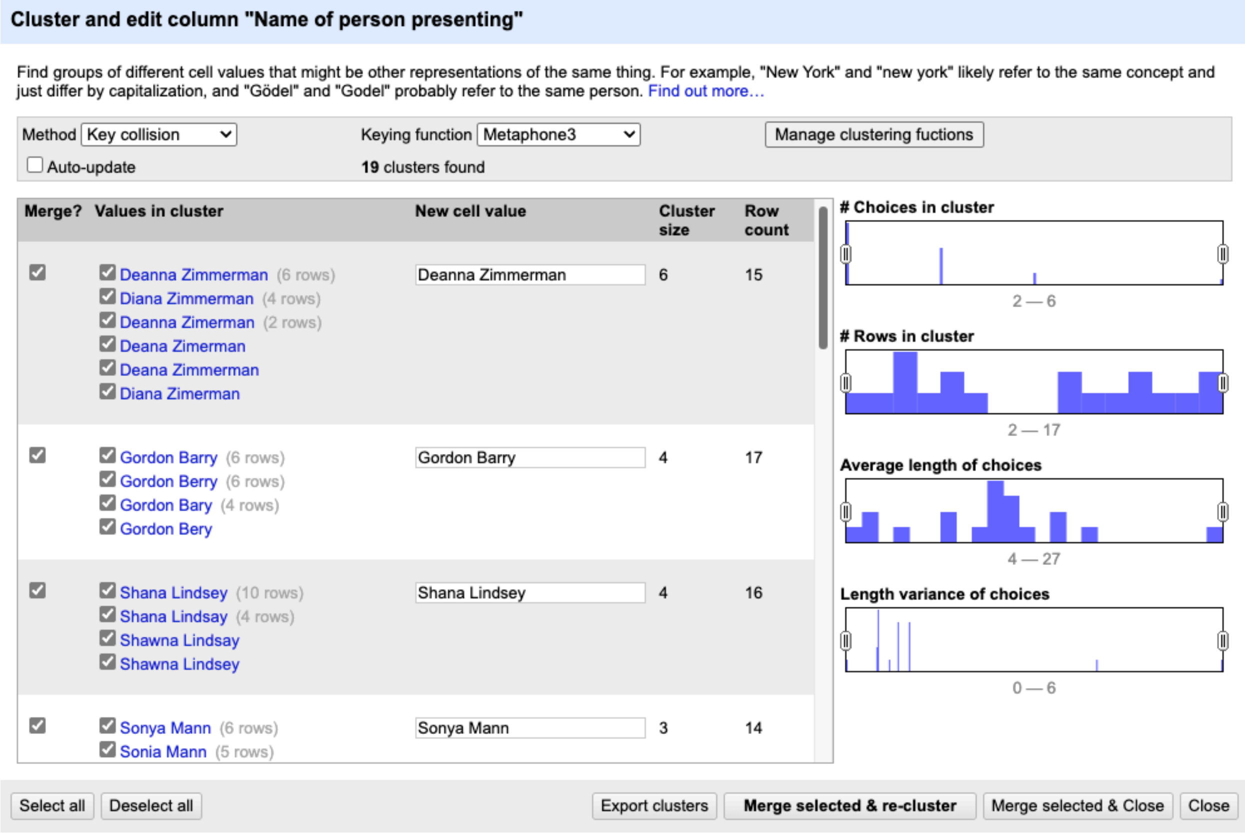
Find groups of different cell values that might be other representations of the same thing. For example, "New York" and "new york" likely refer to the same concept and just differ by capitalization, and "Gödel" and "Godel" probably refer to the same person. Find out more…

Method: Key collision | Keying function: n-Gram fingerprint | n-Gram size: 2

□ Auto-update

3 clusters found

Merge?	Values in cluster	New cell value	Cluster size	Row count
☑	☑ Hattie Garett (2 rows) ☑ Hatie Garett	Hattie Garett	2	3
☑	☑ Sheila Rhodes, Jacob Wheeler (3 rows) ☑ Sheila Rhodes/jacob Wheeler	Sheila Rhodes, Jacob Wheeler	2	4
☑	☑ Hattie Garrett (10 rows) ☑ Hattie Garret (3 rows)	Hattie Garrett	2	13



Cluster and edit column "Name of person presenting"

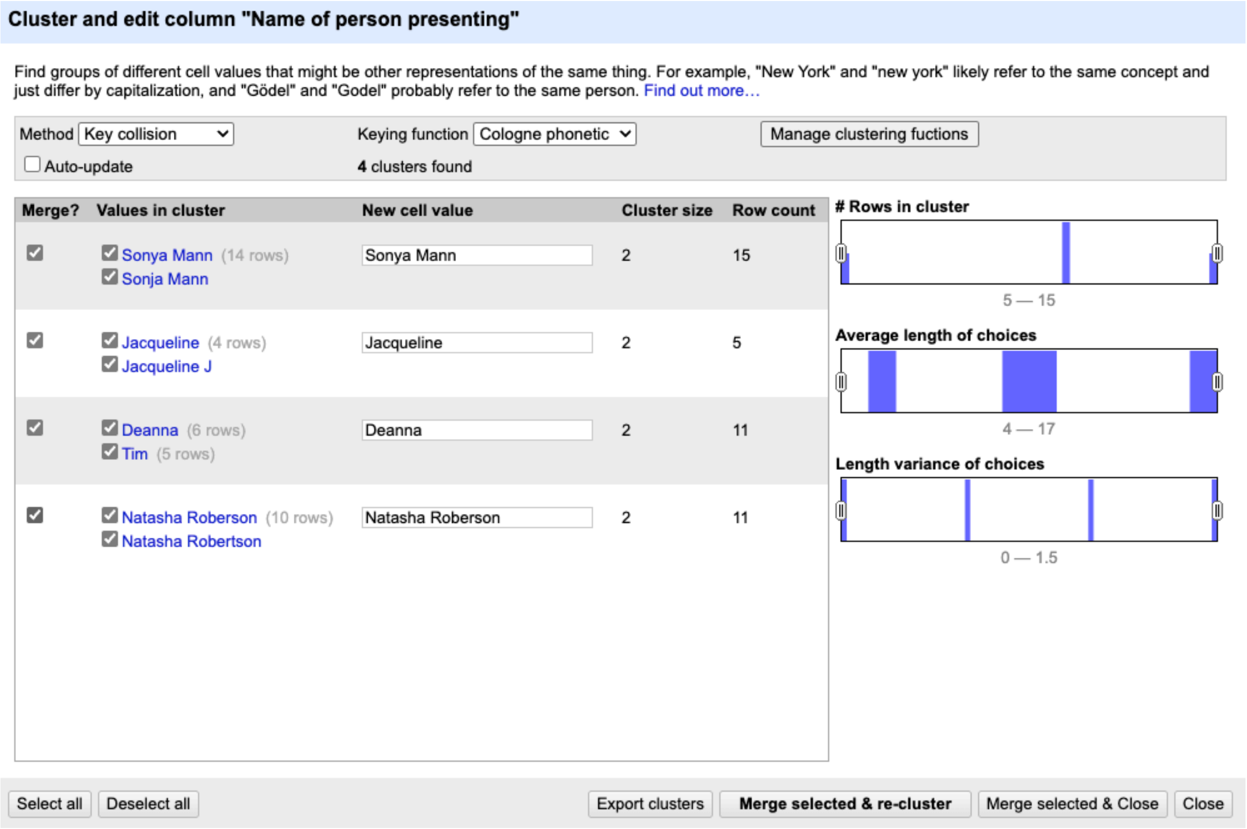
Find groups of different cell values that might be other representations of the same thing. For example, "New York" and "new york" likely refer to the same concept and just differ by capitalization, and "Gödel" and "Godel" probably refer to the same person. Find out more…

Method: Key collision | Keying function: Metaphone3

□ Auto-update

19 clusters found

Merge?	Values in cluster	New cell value	Cluster size	Row count
☑	☑ Deanna Zimmerman (6 rows) ☑ Diana Zimmerman (4 rows) ☑ Deanna Zimerman (2 rows) ☑ Deana Zimerman ☑ Deana Zimmerman ☑ Diana Zimerman	Deanna Zimmerman	6	15
☑	☑ Gordon Barry (6 rows) ☑ Gordon Berry (6 rows) ☑ Gordon Bary (4 rows) ☑ Gordon Bery	Gordon Barry	4	17
☑	☑ Shana Lindsey (10 rows) ☑ Shana Lindsay (4 rows) ☑ Shawna Lindsay ☑ Shawna Lindsey	Shana Lindsey	4	16
☑	☑ Sonya Mann (6 rows) ☑ Sonia Mann (5 rows)	Sonya Mann	3	14



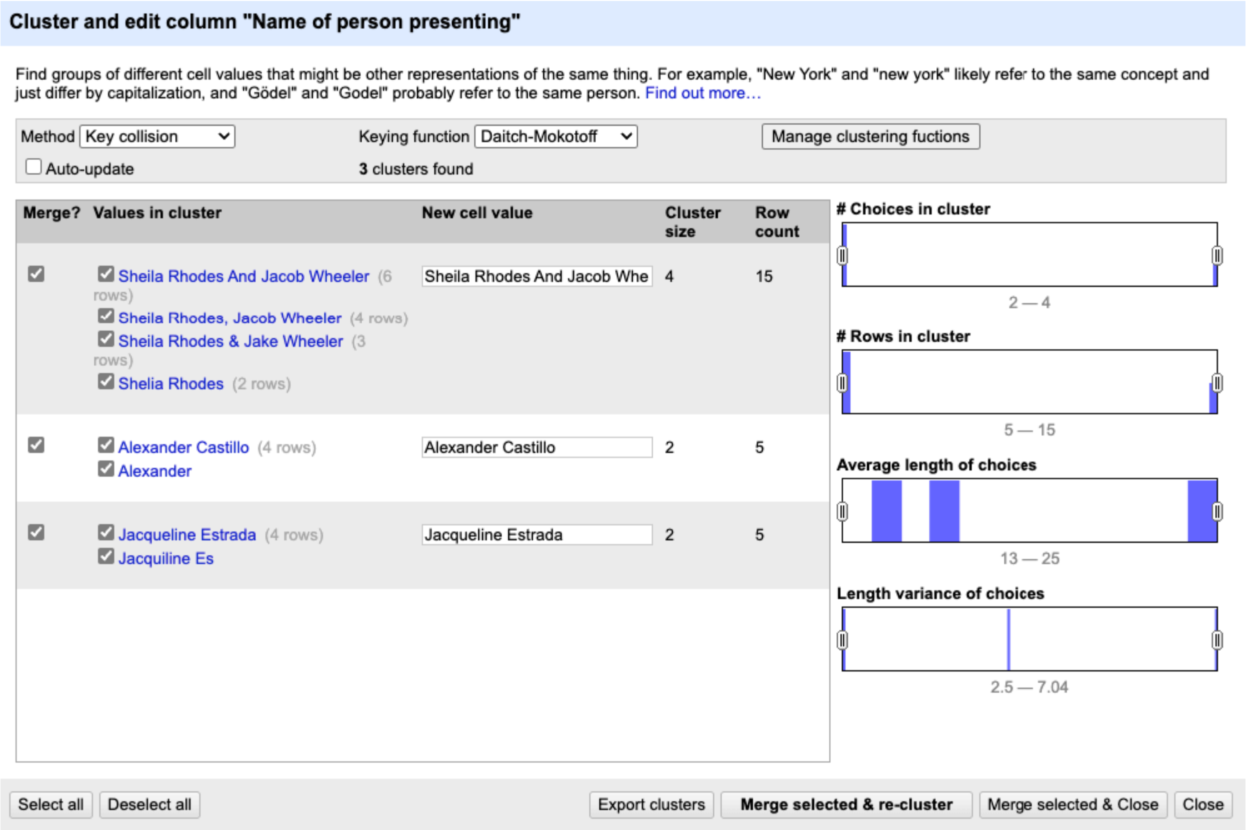
Cluster and edit column "Name of person presenting"

Find groups of different cell values that might be other representations of the same thing. For example, "New York" and "new york" likely refer to the same concept and just differ by capitalization, and "Gödel" and "Godel" probably refer to the same person. Find out more…

Method: Key collision
Keying function: Cologne phonetic
□ Auto-update
4 clusters found

Merge?	Values in cluster	New cell value	Cluster size	Row count
☑	☑ Sonya Mann (14 rows) ☑ Sonja Mann	Sonya Mann	2	15
☑	☑ Jacqueline (4 rows) ☑ Jacqueline J	Jacqueline	2	5
☑	☑ Deanna (6 rows) ☑ Tim (5 rows)	Deanna	2	11
☑	☑ Natasha Roberson (10 rows) ☑ Natasha Robertson	Natasha Roberson	2	11

Manage clustering fuctions
# Rows in cluster
5 — 15
Average length of choices
4 — 17
Length variance of choices
0 — 1.5
Select all
Deselect all
Export clusters
Merge selected & re-cluster
Merge selected & Close
Close



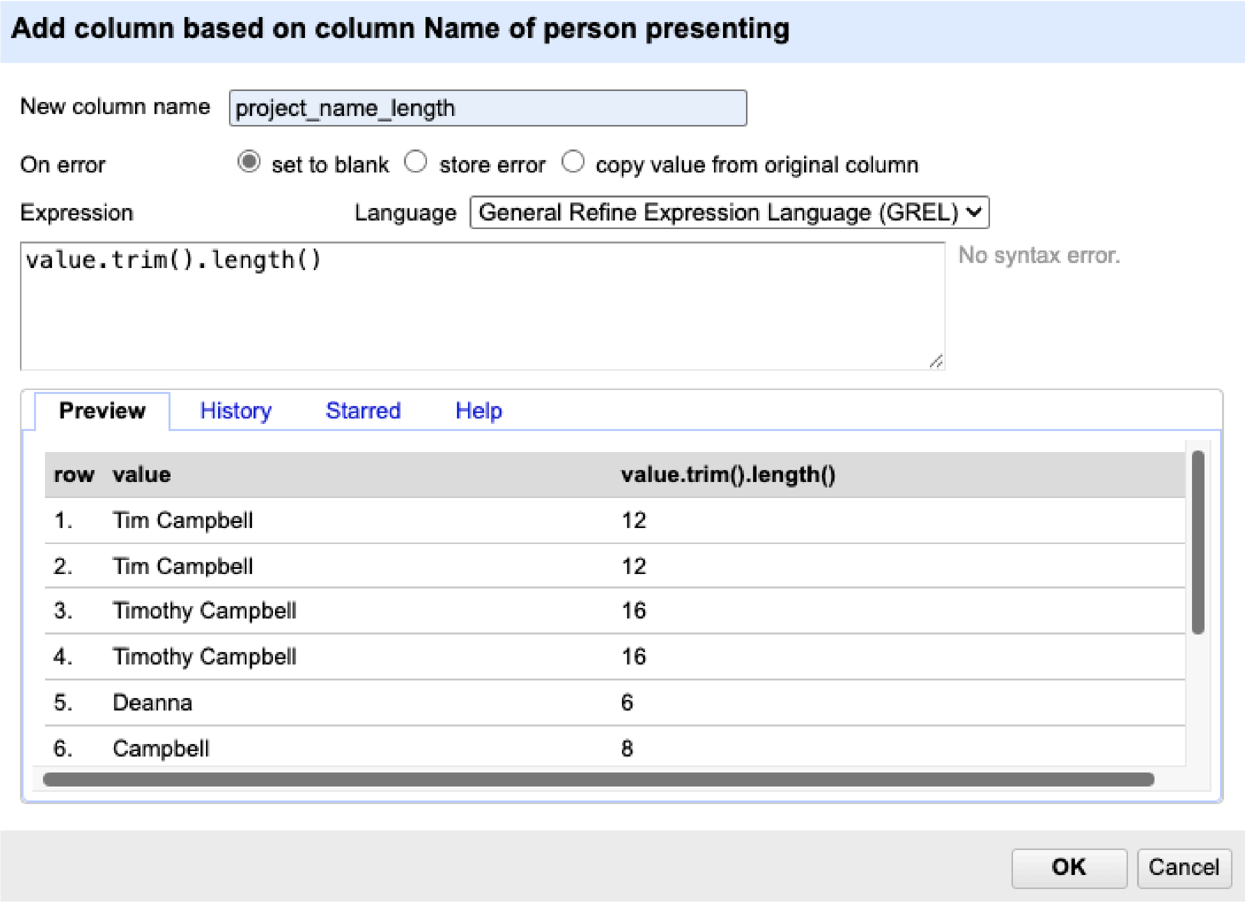
Cluster and edit column "Name of person presenting"

Find groups of different cell values that might be other representations of the same thing. For example, "New York" and "new york" likely refer to the same concept and just differ by capitalization, and "Gödel" and "Godel" probably refer to the same person. Find out more…

Method: Key collision
Keying function: Daitch-Mokotoff
□ Auto-update
3 clusters found

Merge?	Values in cluster	New cell value	Cluster size	Row count
☑	☑ Sheila Rhodes And Jacob Wheeler (6 rows) ☑ Sheila Rhodes, Jacob Wheeler (4 rows) ☑ Sheila Rhodes & Jake Wheeler (3 rows) ☑ Shelia Rhodes (2 rows)	Sheila Rhodes And Jacob Whe	4	15
☑	☑ Alexander Castillo (4 rows) ☑ Alexander	Alexander Castillo	2	5
☑	☑ Jacqueline Estrada (4 rows) ☑ Jacquiline Es	Jacqueline Estrada	2	5

2.In addition to the tutorial perform following manipulations on the dataset provided to you:–Create a new column that contains length of the “Name of project/news package”



Add column based on column Name of person presenting

New column name: project_name_length

On error: ● set to blank ○ store error ○ copy value from original column

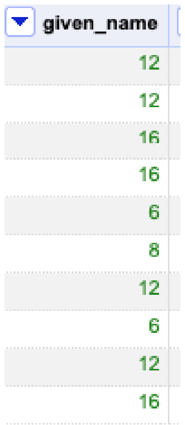
Expression — Language: General Refine Expression Language (GREL)

```
value.trim().length()
```

No syntax error.

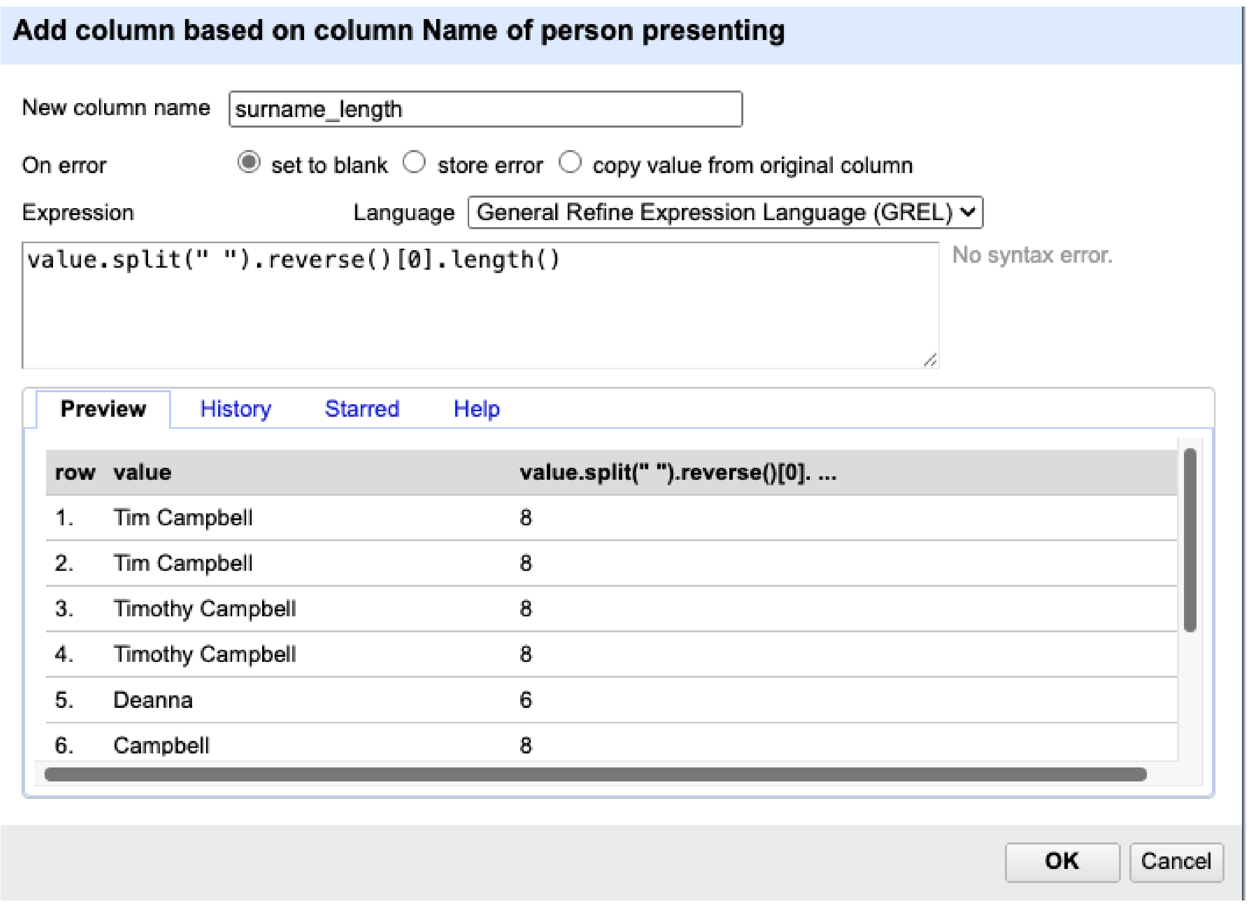
row	value	value.trim().length()
1.	Tim Campbell	12
2.	Tim Campbell	12
3.	Timothy Campbell	16
4.	Timothy Campbell	16
5.	Deanna	6
6.	Campbell	8

–Create a new column that contains length of the “given name”



given_name
12
12
16
16
6
8
12
6
12
16

–Create a new column that contains length of the “surname”



Add column based on column Name of person presenting

New column name: surname_length

On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: General Refine Expression Language (GREL)

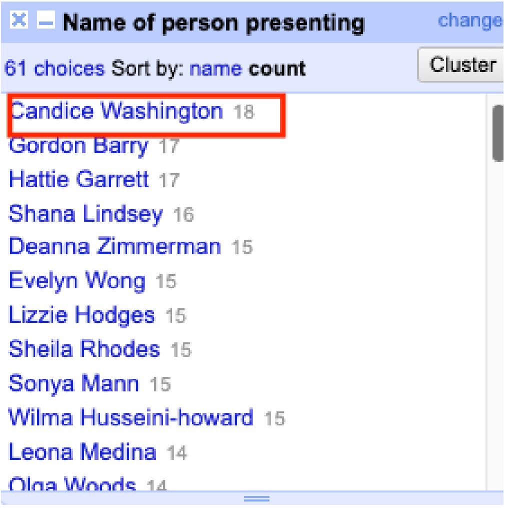
```
value.split(" ").reverse()[0].length()
```

No syntax error.

row	value	value.split(" ").reverse()[0]. ...
1.	Tim Campbell	8
2.	Tim Campbell	8
3.	Timothy Campbell	8
4.	Timothy Campbell	8
5.	Deanna	6
6.	Campbell	8

3.Answer following questions

–What is the longest given name in this dataset?

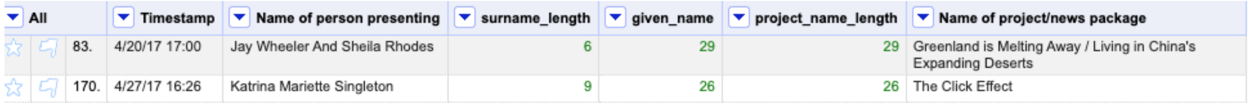


☒ – **Name of person presenting** change

61 choices Sort by: name **count** Cluster

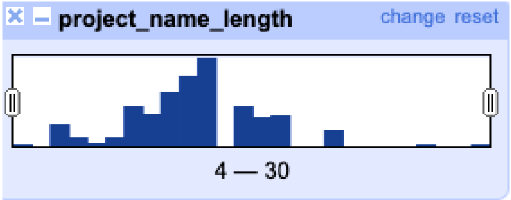
Candice Washington	18
Gordon Barry	17
Hattie Garrett	17
Shana Lindsey	16
Deanna Zimmerman	15
Evelyn Wong	15
Lizzie Hodges	15
Sheila Rhodes	15
Sonya Mann	15
Wilma Husseini-howard	15
Leona Medina	14
Olga Woods	14

–What is the longest "Name of project/news package"



All			Timestamp	Name of person presenting	surname_length	given_name	project_name_length	Name of project/news package
☆		83.	4/20/17 17:00	Jay Wheeler And Sheila Rhodes	6	29	29	Greenland is Melting Away / Living in China's Expanding Deserts
☆		170.	4/27/17 16:26	Katrina Mariette Singleton	9	26	26	The Click Effect

–What is the average length of "Name of project/news package"

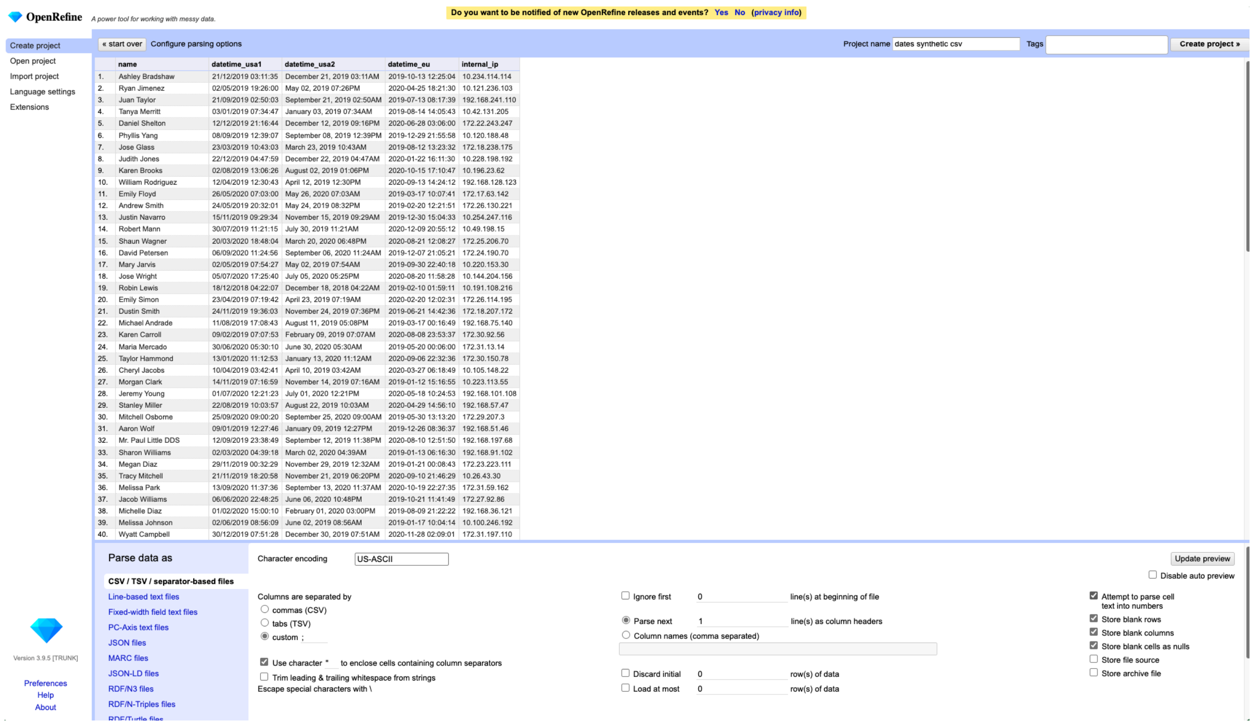


☒ – **project_name_length** change reset

4 — 30

Tasks –Part II

1.Import data from the "dates_synthetic.csv" file into your OpenRefineproject



OpenRefine *A power tool for working with messy data.*

Do you want to be notified of new OpenRefine releases and events? Yes No (privacy info)

Create project
Open project
Import project
Language settings
Extensions

« start over Configure parsing options

Project name: dates synthetic csv Tags: **Create project »**

	name	datetime_usa1	datetime_usa2	datetime_eu	internal_ip
1.	Ashley Bradshaw	21/12/2019 03:11:35	December 21, 2019 03:11AM	2019-10-13 12:25:04	10.234.114.114
2.	Ryan Jimenez	02/05/2019 19:26:00	May 02, 2019 07:26PM	2020-04-25 18:21:30	10.121.236.103
3.	Juan Taylor	21/09/2019 02:50:03	September 21, 2019 02:50AM	2019-07-13 08:17:39	192.168.241.110
4.	Tanya Merritt	03/01/2019 07:34:47	January 03, 2019 07:34AM	2019-08-14 14:05:43	10.42.131.205
5.	Daniel Shelton	12/12/2019 21:16:44	December 12, 2019 09:16PM	2020-06-28 03:06:00	172.22.243.247
6.	Phyllis Yang	08/09/2019 12:39:07	September 08, 2019 12:39PM	2019-12-29 21:55:58	10.120.188.48
7.	Jose Glass	23/03/2019 10:43:03	March 23, 2019 10:43AM	2019-08-12 13:23:32	172.18.238.175
8.	Judith Jones	22/12/2019 04:47:59	December 22, 2019 04:47AM	2020-01-22 16:11:30	10.228.198.192
9.	Karen Brooks	02/08/2019 13:06:26	August 02, 2019 01:06PM	2020-10-15 17:10:47	10.196.23.62
10.	William Rodriguez	12/04/2019 12:30:43	April 12, 2019 12:30PM	2020-09-13 14:24:12	192.168.128.123
11.	Emily Floyd	26/05/2020 07:03:00	May 26, 2020 07:03AM	2019-03-17 10:07:41	172.17.63.142
12.	Andrew Smith	24/05/2019 20:32:01	May 24, 2019 08:32PM	2019-02-20 12:21:51	172.26.130.221
13.	Justin Navarro	15/11/2019 09:29:34	November 15, 2019 09:29AM	2019-12-30 15:04:33	10.254.247.116
14.	Robert Mann	30/07/2019 11:21:15	July 30, 2019 11:21AM	2020-12-09 20:55:12	10.49.198.15
15.	Shaun Wagner	20/03/2020 18:48:04	March 20, 2020 06:48PM	2020-08-21 12:08:27	172.25.206.70
16.	David Petersen	06/09/2020 11:24:56	September 06, 2020 11:24AM	2019-12-07 21:05:21	172.24.190.70
17.	Mary Jarvis	02/05/2019 07:54:27	May 02, 2019 07:54AM	2019-09-30 22:40:18	10.220.153.30
18.	Jose Wright	05/07/2020 17:25:40	July 05, 2020 05:25PM	2020-08-20 11:58:28	10.144.204.156
19.	Robin Lewis	18/12/2018 04:22:07	December 18, 2018 04:22AM	2019-02-10 01:59:11	10.191.108.216
20.	Emily Simon	23/04/2019 07:19:42	April 23, 2019 07:19AM	2020-02-20 12:02:31	172.26.114.195
21.	Dustin Smith	24/11/2019 19:36:03	November 24, 2019 07:36PM	2019-06-21 14:42:36	172.18.207.172
22.	Michael Andrade	11/08/2019 17:08:43	August 11, 2019 05:08PM	2019-03-17 00:16:49	192.168.75.140
23.	Karen Carroll	09/02/2019 07:07:53	February 09, 2019 07:07AM	2020-08-08 23:53:37	172.30.92.56
24.	Maria Mercado	30/06/2020 05:30:10	June 30, 2020 05:30AM	2019-05-20 00:06:00	172.31.13.14
25.	Taylor Hammond	13/01/2020 11:12:53	January 13, 2020 11:12AM	2020-09-06 22:32:36	172.30.150.78
26.	Cheryl Jacobs	10/04/2019 03:42:41	April 10, 2019 03:42AM	2020-03-27 06:18:49	10.105.148.22
27.	Morgan Clark	14/11/2019 07:16:59	November 14, 2019 07:16AM	2019-01-12 15:16:55	10.223.113.55
28.	Jeremy Young	01/07/2020 12:21:23	July 01, 2020 12:21PM	2020-05-18 10:24:53	192.168.101.108
29.	Stanley Miller	22/08/2019 10:03:57	August 22, 2019 10:03AM	2020-04-29 14:56:10	192.168.57.47
30.	Mitchell Osborne	25/09/2020 09:00:20	September 25, 2020 09:00AM	2019-05-30 13:13:20	172.29.207.3
31.	Aaron Wolf	09/01/2019 12:27:46	January 09, 2019 12:27PM	2019-12-26 08:36:37	192.168.51.46
32.	Mr. Paul Little DDS	12/09/2019 23:38:49	September 12, 2019 11:38PM	2020-08-10 12:51:50	192.168.197.68
33.	Sharon Williams	02/03/2020 04:39:18	March 02, 2020 04:39AM	2019-01-13 06:16:30	192.168.91.102
34.	Megan Diaz	29/11/2019 00:32:29	November 29, 2019 12:32AM	2019-01-21 00:08:43	172.23.223.111
35.	Tracy Mitchell	21/11/2019 18:20:58	November 21, 2019 06:20PM	2020-09-10 21:46:29	10.26.43.30
36.	Melissa Park	13/09/2020 11:37:36	September 13, 2020 11:37AM	2020-10-19 22:27:35	172.31.59.162
37.	Jacob Williams	06/06/2020 22:48:25	June 06, 2020 10:48PM	2019-10-21 11:41:49	172.27.92.86
38.	Michelle Diaz	01/02/2020 15:00:10	February 01, 2020 03:00PM	2019-08-09 21:22:22	192.168.36.121
39.	Melissa Johnson	02/06/2019 08:56:09	June 02, 2019 08:56AM	2019-01-17 10:04:14	10.100.246.192
40.	Wyatt Campbell	30/12/2019 07:51:28	December 30, 2019 07:51AM	2020-11-28 02:09:01	172.31.197.110

Parse data as

CSV / TSV / separator-based files
Line-based text files
Fixed-width field text files
PC-Axis text files
JSON files
MARC files
JSON-LD files
RDF/N3 files
RDF/N-Triples files

Version 3.9.5 [TRUNK]

Preferences
Help
About

Character encoding: US-ASCII

Update preview
□ Disable auto preview

Columns are separated by
○ commas (CSV)
○ tabs (TSV)
● custom ;

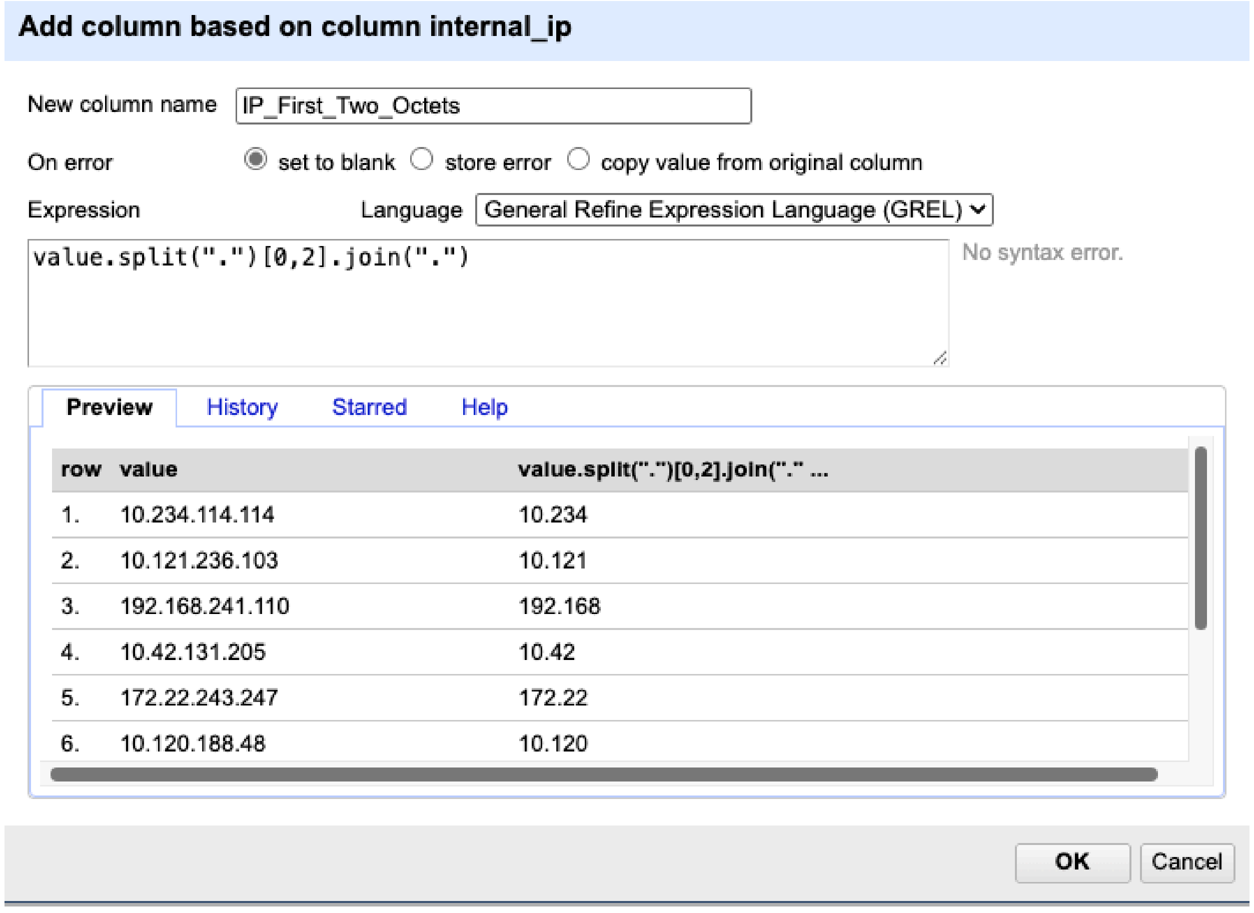
☑ Use character " to enclose cells containing column separators
□ Trim leading & trailing whitespace from strings
Escape special characters with \

□ Ignore first 0 line(s) at beginning of file
● Parse next 1 line(s) as column headers
○ Column names (comma separated)

□ Discard initial 0 row(s) of data
□ Load at most 0 row(s) of data

☑ Attempt to parse cell text into numbers
☑ Store blank rows
☑ Store blank columns
☑ Store blank cells as nulls
□ Store file source
□ Store archive file

2.Create a new column that contains the first 2 octets of the provided IP address



Add column based on column internal_ip

New column name: IP_First_Two_Octets

On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: General Refine Expression Language (GREL)

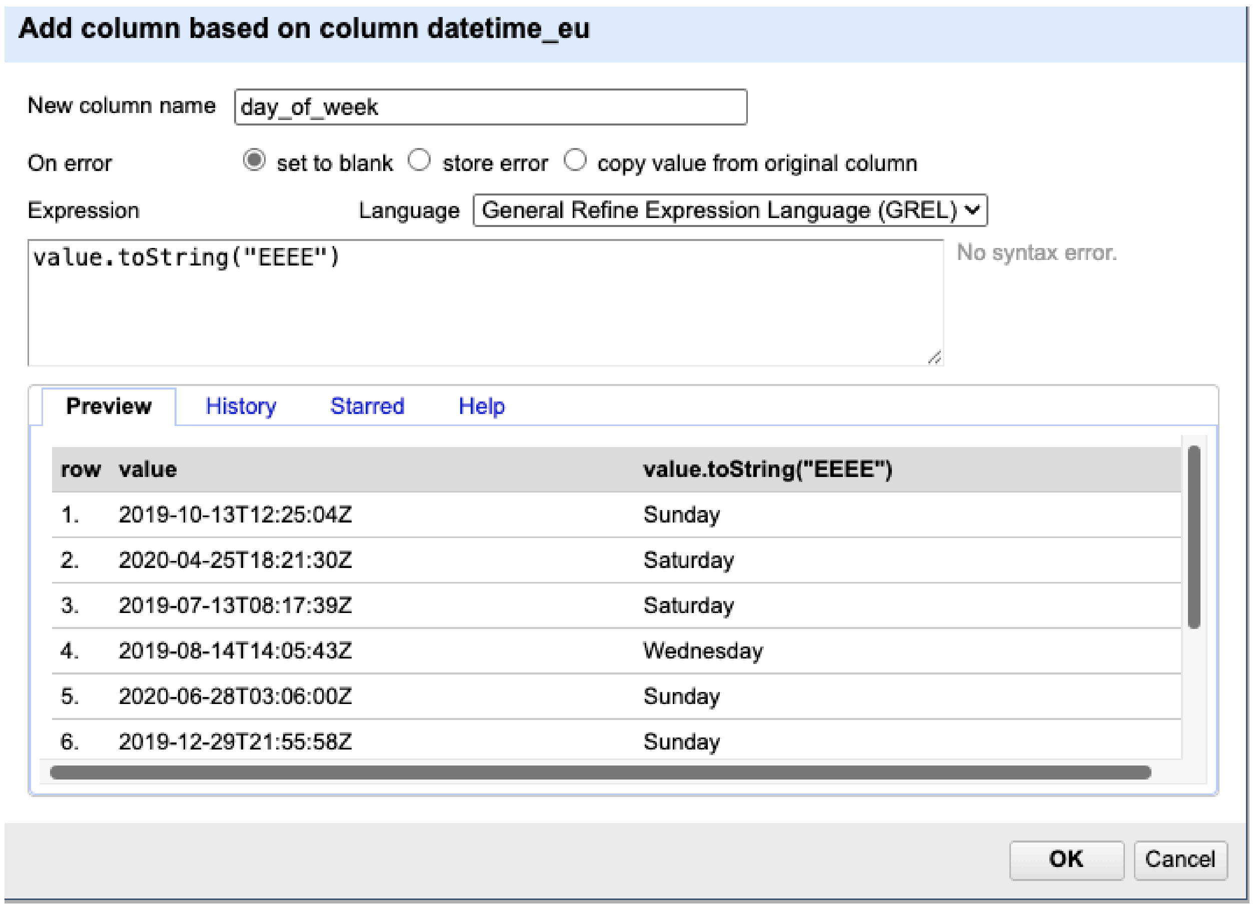
```
value.split(".")[0,2].join(".")
```

No syntax error.

row	value	value.split(".")[0,2].join("." ...
1.	10.234.114.114	10.234
2.	10.121.236.103	10.121
3.	192.168.241.110	192.168
4.	10.42.131.205	10.42
5.	172.22.243.247	172.22
6.	10.120.188.48	10.120

3.Create multiple new columns with the following content for each date in the column “datetime_eu”

–day of the week



Add column based on column datetime_eu

New column name: day_of_week

On error: ● set to blank ○ store error ○ copy value from original column

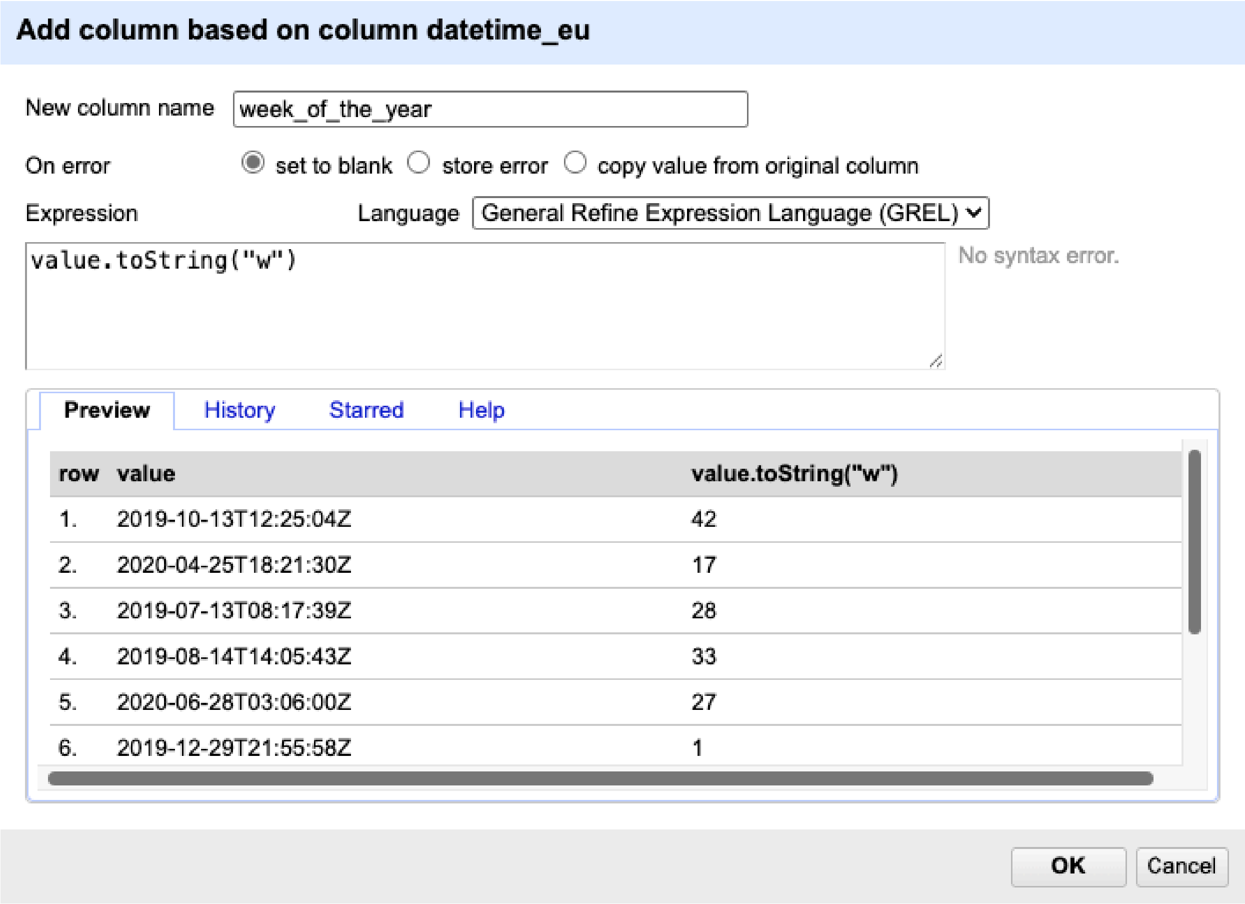
Expression — Language: General Refine Expression Language (GREL)

```
value.toString("EEEE")
```

No syntax error.

row	value	value.toString("EEEE")
1.	2019-10-13T12:25:04Z	Sunday
2.	2020-04-25T18:21:30Z	Saturday
3.	2019-07-13T08:17:39Z	Saturday
4.	2019-08-14T14:05:43Z	Wednesday
5.	2020-06-28T03:06:00Z	Sunday
6.	2019-12-29T21:55:58Z	Sunday

–week of the year



Add column based on column datetime_eu

New column name: week_of_the_year

On error: ● set to blank ○ store error ○ copy value from original column

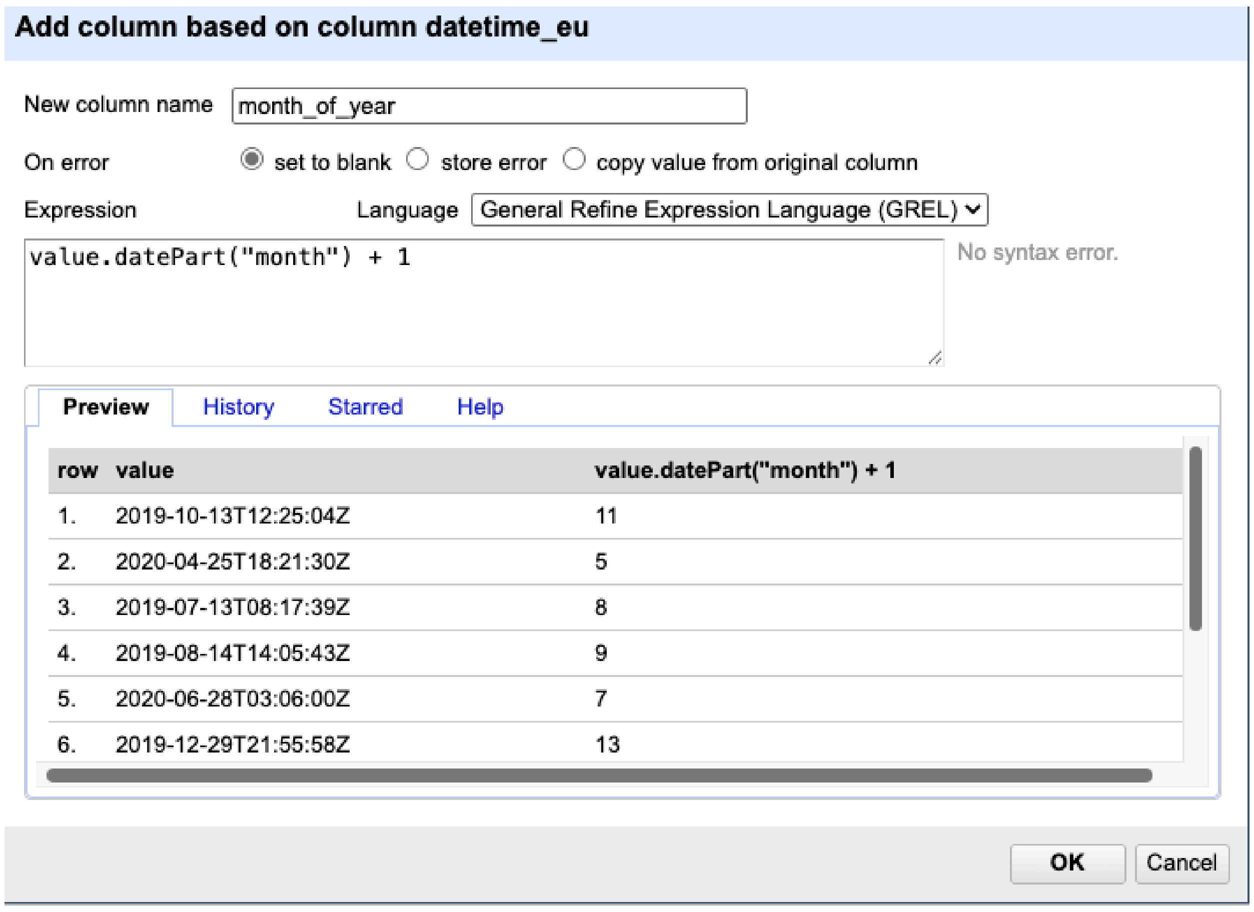
Expression — Language: General Refine Expression Language (GREL)

```
value.toString("w")
```

No syntax error.

row	value	value.toString("w")
1.	2019-10-13T12:25:04Z	42
2.	2020-04-25T18:21:30Z	17
3.	2019-07-13T08:17:39Z	28
4.	2019-08-14T14:05:43Z	33
5.	2020-06-28T03:06:00Z	27
6.	2019-12-29T21:55:58Z	1

–month of the year (as number)



Add column based on column datetime_eu

New column name: month_of_year

On error: ● set to blank ○ store error ○ copy value from original column

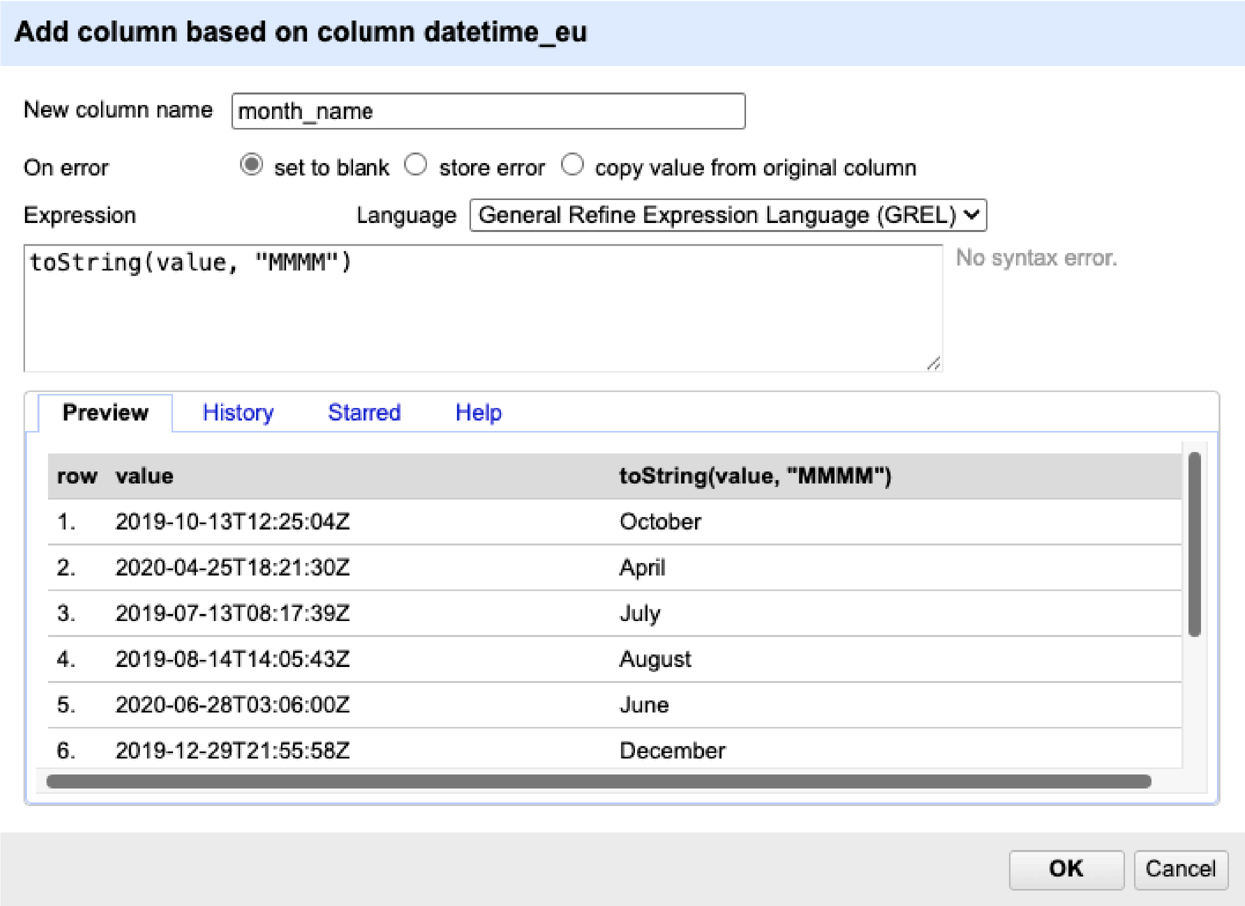
Expression — Language: General Refine Expression Language (GREL)

```
value.datePart("month") + 1
```

No syntax error.

row	value	value.datePart("month") + 1
1.	2019-10-13T12:25:04Z	11
2.	2020-04-25T18:21:30Z	5
3.	2019-07-13T08:17:39Z	8
4.	2019-08-14T14:05:43Z	9
5.	2020-06-28T03:06:00Z	7
6.	2019-12-29T21:55:58Z	13

–month of the year (as string -'January')



Add column based on column datetime_eu

New column name: month_name

On error: ● set to blank ○ store error ○ copy value from original column

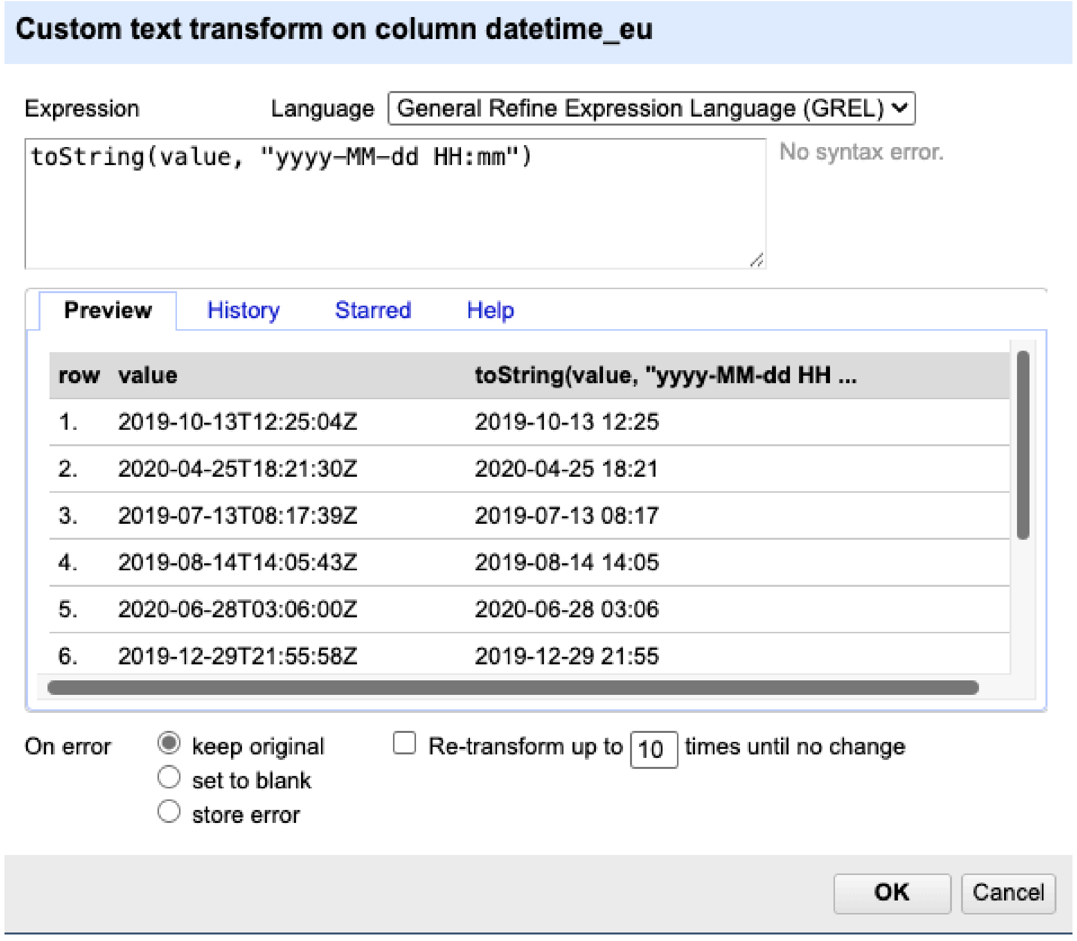
Expression — Language: General Refine Expression Language (GREL)

```
toString(value, "MMMM")
```

No syntax error.

row	value	toString(value, "MMMM")
1.	2019-10-13T12:25:04Z	October
2.	2020-04-25T18:21:30Z	April
3.	2019-07-13T08:17:39Z	July
4.	2019-08-14T14:05:43Z	August
5.	2020-06-28T03:06:00Z	June
6.	2019-12-29T21:55:58Z	December

4.Homogenize given date and time for all columns with datetime value, use the following format "%Y-%m-%d %H:%M"



Custom text transform on column datetime_eu

Expression — Language: General Refine Expression Language (GREL)

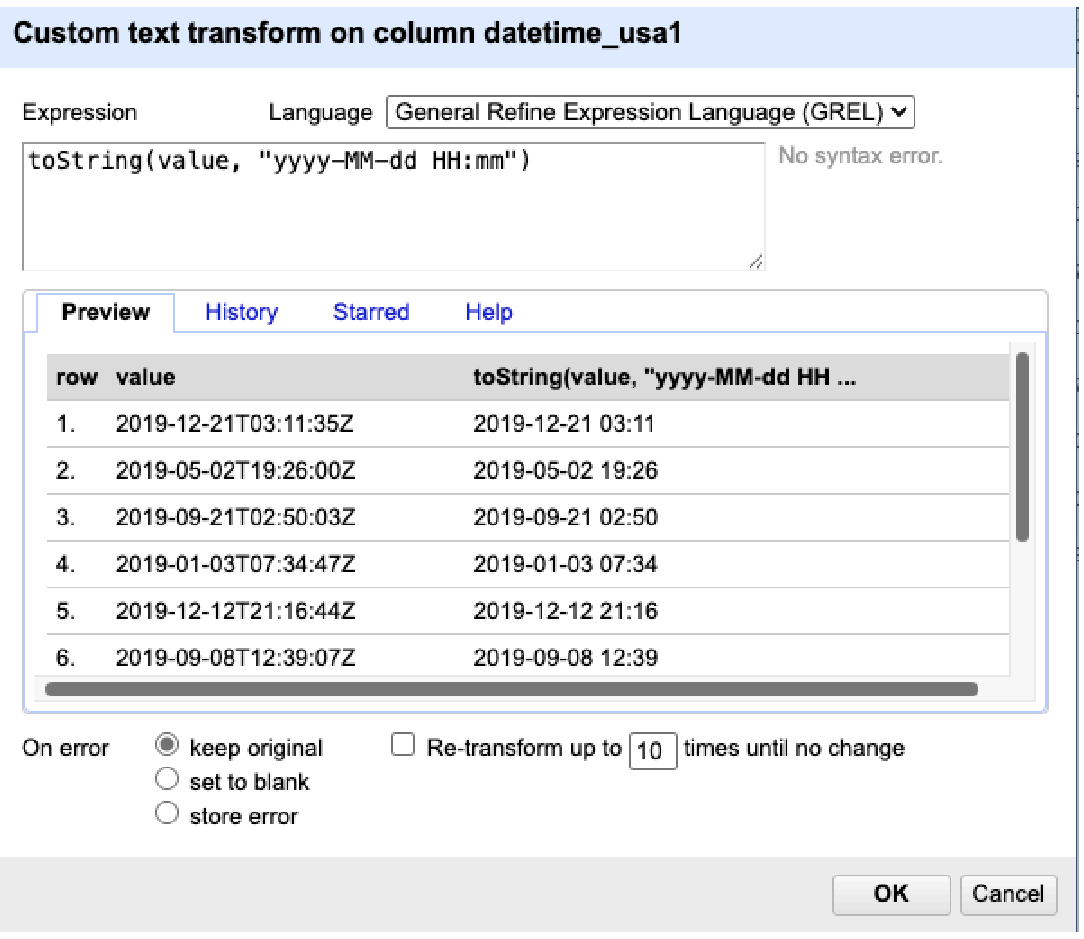
```
toString(value, "yyyy-MM-dd HH:mm")
```

No syntax error.

row	value	toString(value, "yyyy-MM-dd HH ...
1.	2019-10-13T12:25:04Z	2019-10-13 12:25
2.	2020-04-25T18:21:30Z	2020-04-25 18:21
3.	2019-07-13T08:17:39Z	2019-07-13 08:17
4.	2019-08-14T14:05:43Z	2019-08-14 14:05
5.	2020-06-28T03:06:00Z	2020-06-28 03:06
6.	2019-12-29T21:55:58Z	2019-12-29 21:55

On error: ● keep original ○ set to blank ○ store error

□ Re-transform up to 10 times until no change



Custom text transform on column datetime_usa1

Expression Language General Refine Expression Language (GREL)

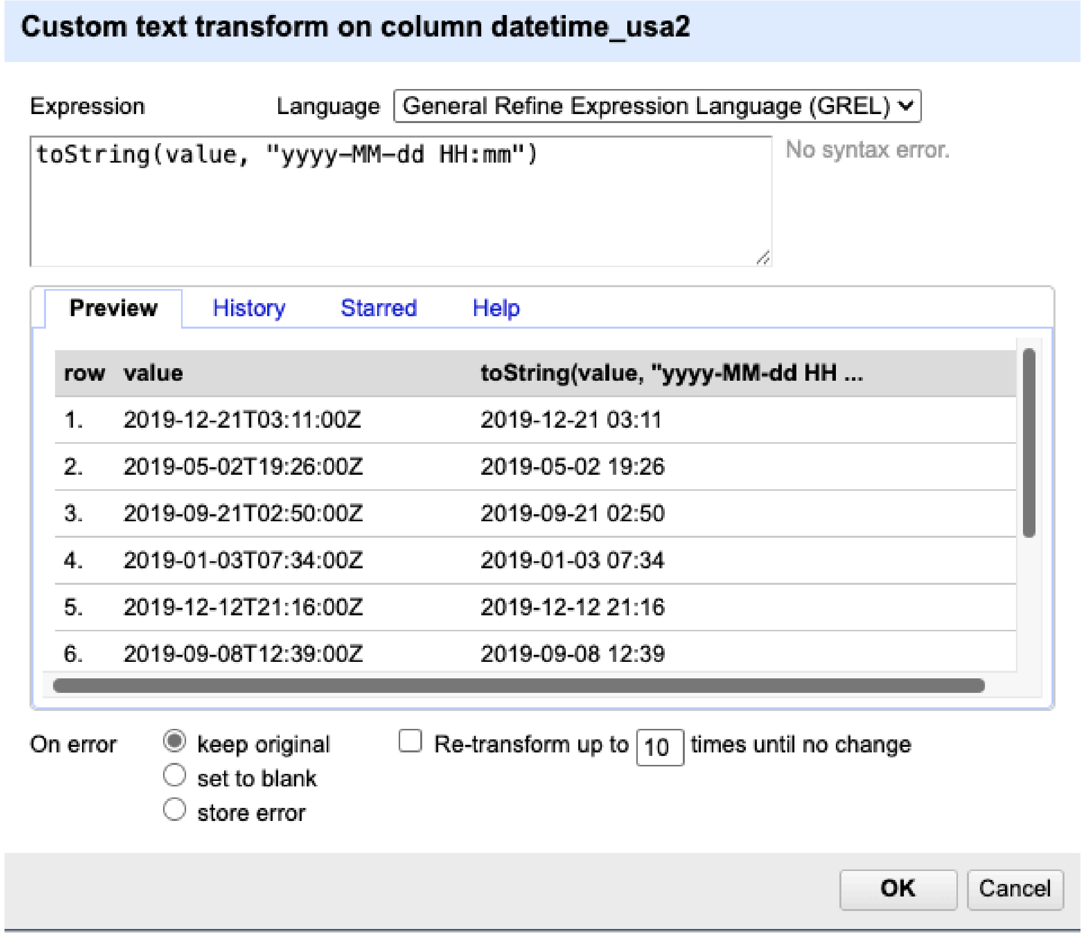
```
toString(value, "yyyy-MM-dd HH:mm")
```

No syntax error.

row	value	toString(value, "yyyy-MM-dd HH ...
1.	2019-12-21T03:11:35Z	2019-12-21 03:11
2.	2019-05-02T19:26:00Z	2019-05-02 19:26
3.	2019-09-21T02:50:03Z	2019-09-21 02:50
4.	2019-01-03T07:34:47Z	2019-01-03 07:34
5.	2019-12-12T21:16:44Z	2019-12-12 21:16
6.	2019-09-08T12:39:07Z	2019-09-08 12:39

On error ● keep original ○ set to blank ○ store error

□ Re-transform up to 10 times until no change



Custom text transform on column datetime_usa2

Expression Language General Refine Expression Language (GREL)

```
toString(value, "yyyy-MM-dd HH:mm")
```

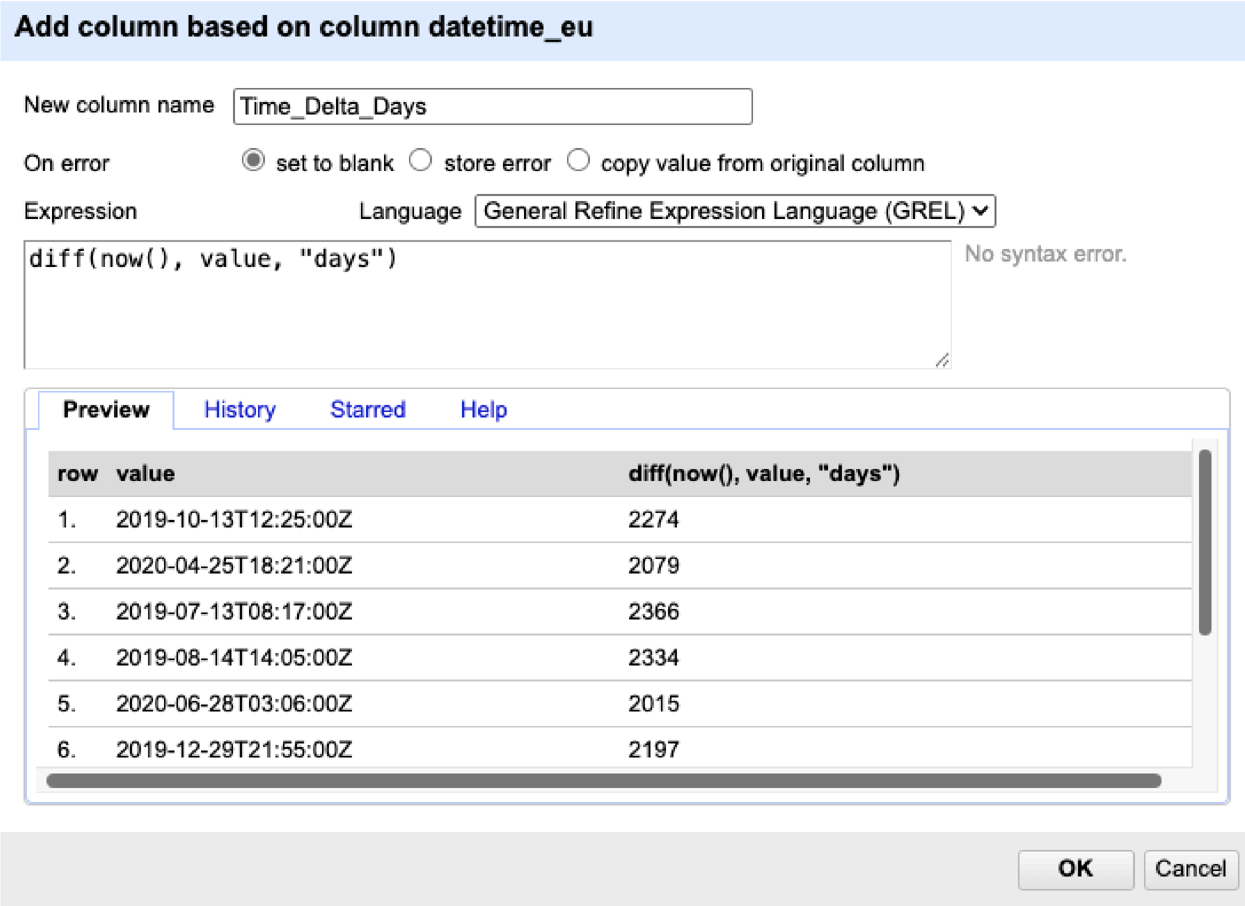
No syntax error.

row	value	toString(value, "yyyy-MM-dd HH ...
1.	2019-12-21T03:11:00Z	2019-12-21 03:11
2.	2019-05-02T19:26:00Z	2019-05-02 19:26
3.	2019-09-21T02:50:00Z	2019-09-21 02:50
4.	2019-01-03T07:34:00Z	2019-01-03 07:34
5.	2019-12-12T21:16:00Z	2019-12-12 21:16
6.	2019-09-08T12:39:00Z	2019-09-08 12:39

On error ● keep original ○ set to blank ○ store error

□ Re-transform up to 10 times until no change

5.Calculate “time delta” with today's date for the column with the date "datetime_eu”



Add column based on column datetime_eu

New column name: Time_Delta_Days

On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: General Refine Expression Language (GREL)

```
diff(now(), value, "days")
```

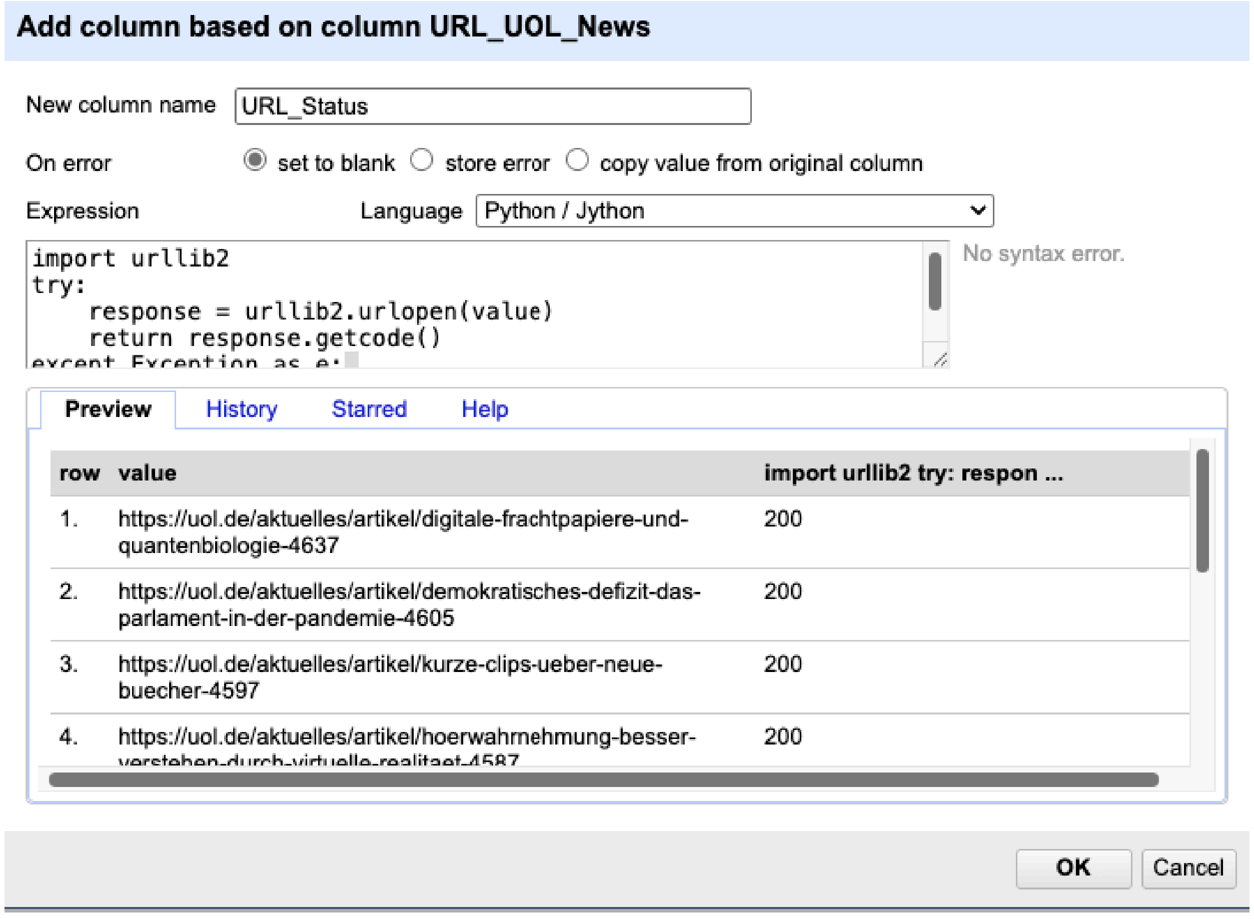
No syntax error.

row	value	diff(now(), value, "days")
1.	2019-10-13T12:25:00Z	2274
2.	2020-04-25T18:21:00Z	2079
3.	2019-07-13T08:17:00Z	2366
4.	2019-08-14T14:05:00Z	2334
5.	2020-06-28T03:06:00Z	2015
6.	2019-12-29T21:55:00Z	2197

Tasks –Part III

1.Import data from the "UOL News.csv" file into your OpenRefineproject

2.Check, if provided within the dataset URLs are still available



Add column based on column URL_UOL_News

New column name: URL_Status

On error: ● set to blank ○ store error ○ copy value from original column

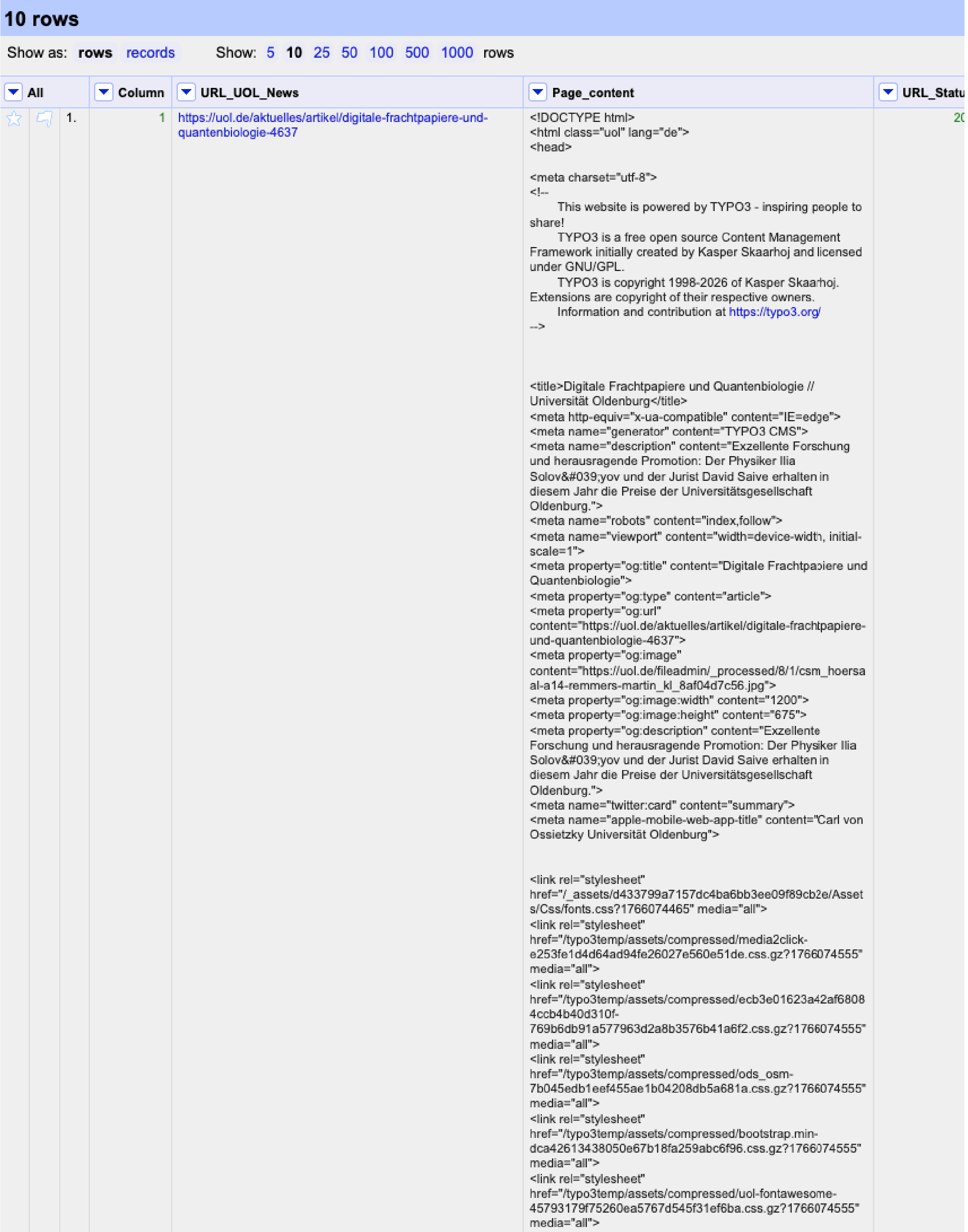
Expression — Language: Python / Jython

```
import urllib2
try:
    response = urllib2.urlopen(value)
    return response.getcode()
except Exception as e:
```

No syntax error.

row	value	import urllib2 try: respon ...
1.	https://uol.de/aktuelles/artikel/digitale-frachtpapiere-und-quantenbiologie-4637	200
2.	https://uol.de/aktuelles/artikel/demokratisches-defizit-das-parlament-in-der-pandemie-4605	200
3.	https://uol.de/aktuelles/artikel/kurze-clips-ueber-neue-buecher-4597	200
4.	https://uol.de/aktuelles/artikel/hoerwahrnehmung-besser-verstehen-durch-virtuelle-realitaet-4587	200

3.Fetch data from a URL using "Add Column By Fetching URL“ feature



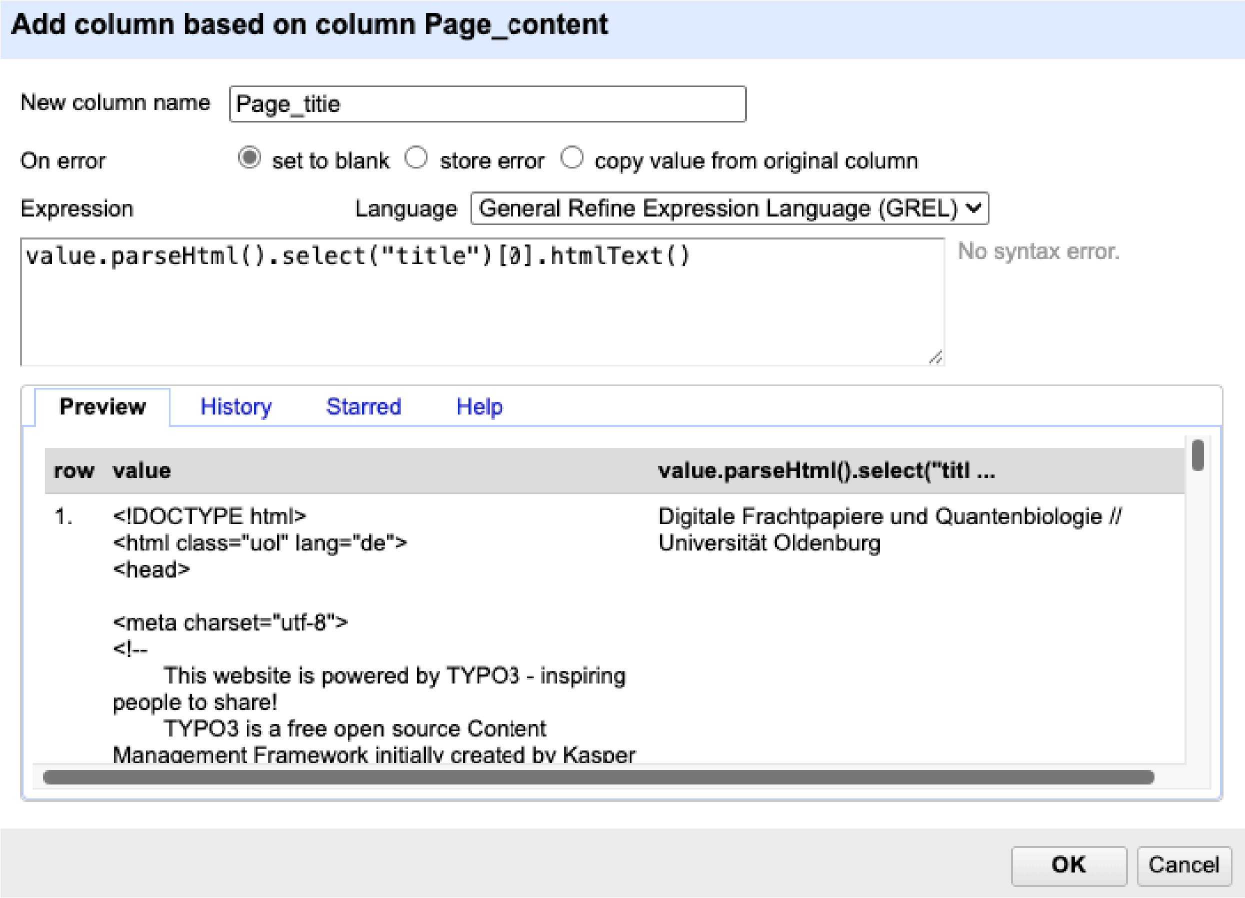
10 rows

Show as: **rows** records Show: 5 **10** 25 50 100 500 1000 rows

All	Column	URL_UOL_News	Page_content	URL_Statu
☆ 1.	1	https://uol.de/aktuelles/artikel/digitale-frachtpapiere-und-quantenbiologie-4637	<!DOCTYPE html> <html class="uol" lang="de"> <head> <meta charset="utf-8"> <!-- This website is powered by TYPO3 - inspiring people to share! TYPO3 is a free open source Content Management Framework initially created by Kasper Skaarhoj and licensed under GNU/GPL. TYPO3 is copyright 1998-2026 of Kasper Skaarhoj. Extensions are copyright of their respective owners. Information and contribution at https://typo3.org/ --> <title>Digitale Frachtpapiere und Quantenbiologie // Universität Oldenburg</title> <meta http-equiv="x-ua-compatible" content="IE=edge"> <meta name="generator" content="TYPO3 CMS"> <meta name="description" content="Exzellente Forschung und herausragende Promotion: Der Physiker Ilia Solov&#039;yov und der Jurist David Saive erhalten in diesem Jahr die Preise der Universitätsgesellschaft Oldenburg."> <meta name="robots" content="index,follow"> <meta name="viewport" content="width=device-width, initial-scale=1"> <meta property="og:title" content="Digitale Frachtpapiere und Quantenbiologie"> <meta property="og:type" content="article"> <meta property="og:url" content="https://uol.de/aktuelles/artikel/digitale-frachtpapiere-und-quantenbiologie-4637"> <meta property="og:image" content="https://uol.de/fileadmin/_processed/8/1/csm_hoersaal-a14-remmers-martin_kl_8af04d7c56.jpg"> <meta property="og:image:width" content="1200"> <meta property="og:image:height" content="675"> <meta property="og:description" content="Exzellente Forschung und herausragende Promotion: Der Physiker Ilia Solov&#039;yov und der Jurist David Saive erhalten in diesem Jahr die Preise der Universitätsgesellschaft Oldenburg."> <meta name="twitter:card" content="summary"> <meta name="apple-mobile-web-app-title" content="Carl von Ossietzky Universität Oldenburg"> <link rel="stylesheet" href="/_assets/d433799a7157dc4ba6bb3ee09f89cb2e/Assets/Css/fonts.css?1766074465" media="all"> <link rel="stylesheet" href="/typo3temp/assets/compressed/media2click-e253fe1d4d64ad94fe26027e560e51de.css.gz?1766074555" media="all"> <link rel="stylesheet" href="/typo3temp/assets/compressed/ecb3e01623a42af68084ccb4b40d310f-769b6db91a577963d2a8b3576b41a6f2.css.gz?1766074555" media="all"> <link rel="stylesheet" href="/typo3temp/assets/compressed/ods_osm-7b045edb1eef455ae1b04208db5a681a.css.gz?1766074555" media="all"> <link rel="stylesheet" href="/typo3temp/assets/compressed/bootstrap.min-dca42613438050e67b18fa259abc6f96.css.gz?1766074555" media="all"> <link rel="stylesheet" href="/typo3temp/assets/compressed/uol-fontawesome-45793179f75260ea5767d545f31ef6ba.css.gz?1766074555" media="all">	20

4.For the each "item" below, create a separate column based on the downloaded content for each URL:

–page title



Add column based on column Page_content

New column name: Page_titie

On error: ● set to blank ○ store error ○ copy value from original column

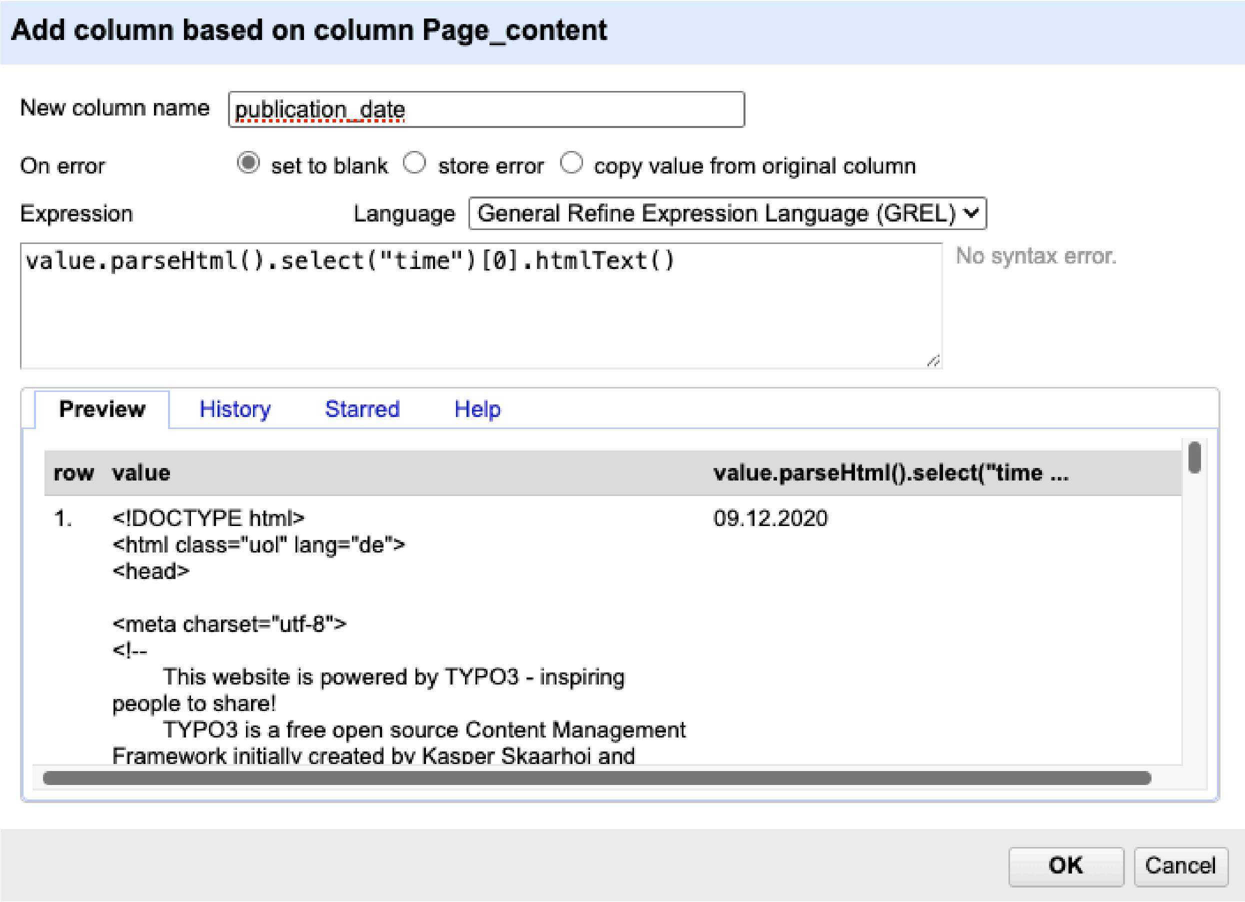
Expression Language: General Refine Expression Language (GREL)

```
value.parseHtml().select("title")[0].htmlText()
```

No syntax error.

row	value	value.parseHtml().select("titl ...
1.	<!DOCTYPE html> <html class="uol" lang="de"> <head> <meta charset="utf-8"> <!-- This website is powered by TYPO3 - inspiring people to share! TYPO3 is a free open source Content Management Framework initially created by Kasper	Digitale Frachtpapiere und Quantenbiologie // Universität Oldenburg

–publication date



Add column based on column Page_content

New column name: publication_date

On error: ● set to blank ○ store error ○ copy value from original column

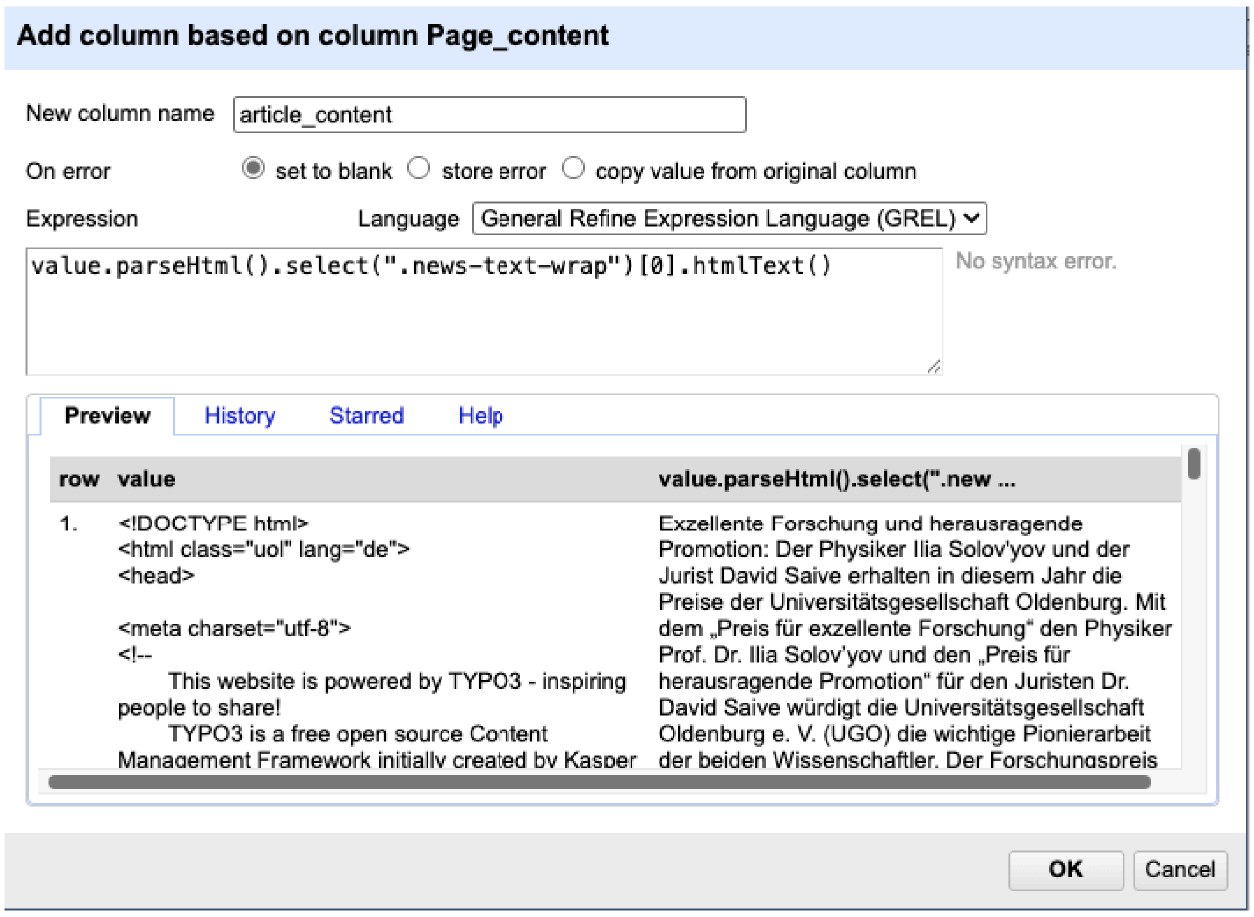
Expression — Language: General Refine Expression Language (GREL)

```
value.parseHtml().select("time")[0].htmlText()
```

No syntax error.

row	value	value.parseHtml().select("time ...
1.	<!DOCTYPE html> <html class="uol" lang="de"> <head> <meta charset="utf-8"> <!-- This website is powered by TYPO3 - inspiring people to share! TYPO3 is a free open source Content Management Framework initially created by Kasper Skaarhoj and	09.12.2020

–content of the main article for each given URL



Add column based on column Page_content

New column name: article_content

On error: ● set to blank ○ store error ○ copy value from original column

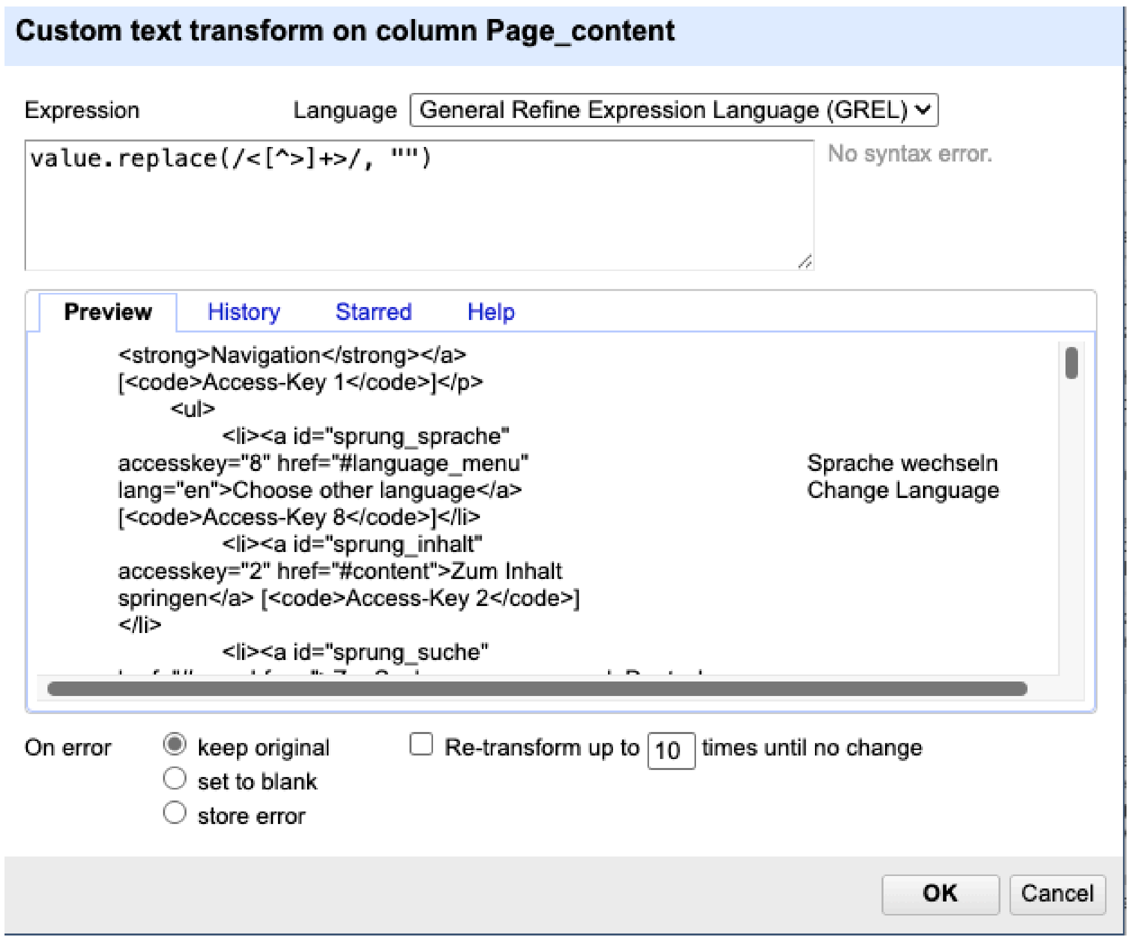
Expression — Language: General Refine Expression Language (GREL)

```
value.parseHtml().select(".news-text-wrap")[0].htmlText()
```

No syntax error.

row	value	value.parseHtml().select(".new ...
1.	<!DOCTYPE html> <html class="uol" lang="de"> <head> <meta charset="utf-8"> <!-- This website is powered by TYPO3 - inspiring people to share! TYPO3 is a free open source Content Management Framework initially created by Kasper	Exzellente Forschung und herausragende Promotion: Der Physiker Ilia Solov'yov und der Jurist David Saive erhalten in diesem Jahr die Preise der Universitätsgesellschaft Oldenburg. Mit dem „Preis für exzellente Forschung“ den Physiker Prof. Dr. Ilia Solov'yov und den „Preis für herausragende Promotion“ für den Juristen Dr. David Saive würdigt die Universitätsgesellschaft Oldenburg e. V. (UGO) die wichtige Pionierarbeit der beiden Wissenschaftler. Der Forschungspreis

–*(optional -remove HTML tags)



Custom text transform on column Page_content

Expression Language: General Refine Expression Language (GREL)

```
value.replace(/<[^>]+>/, "")
```

No syntax error.

```
        <strong>Navigation</strong></a>
        [<code>Access-Key 1</code>]</p>
              <ul>
                    <li><a id="sprung_sprache"
        accesskey="8" href="#language_menu"                  Sprache wechseln
        lang="en">Choose other language</a>                  Change Language
        [<code>Access-Key 8</code>]</li>
                    <li><a id="sprung_inhalt"
        accesskey="2" href="#content">Zum Inhalt
        springen</a> [<code>Access-Key 2</code>]
        </li>
                    <li><a id="sprung_suche"
```

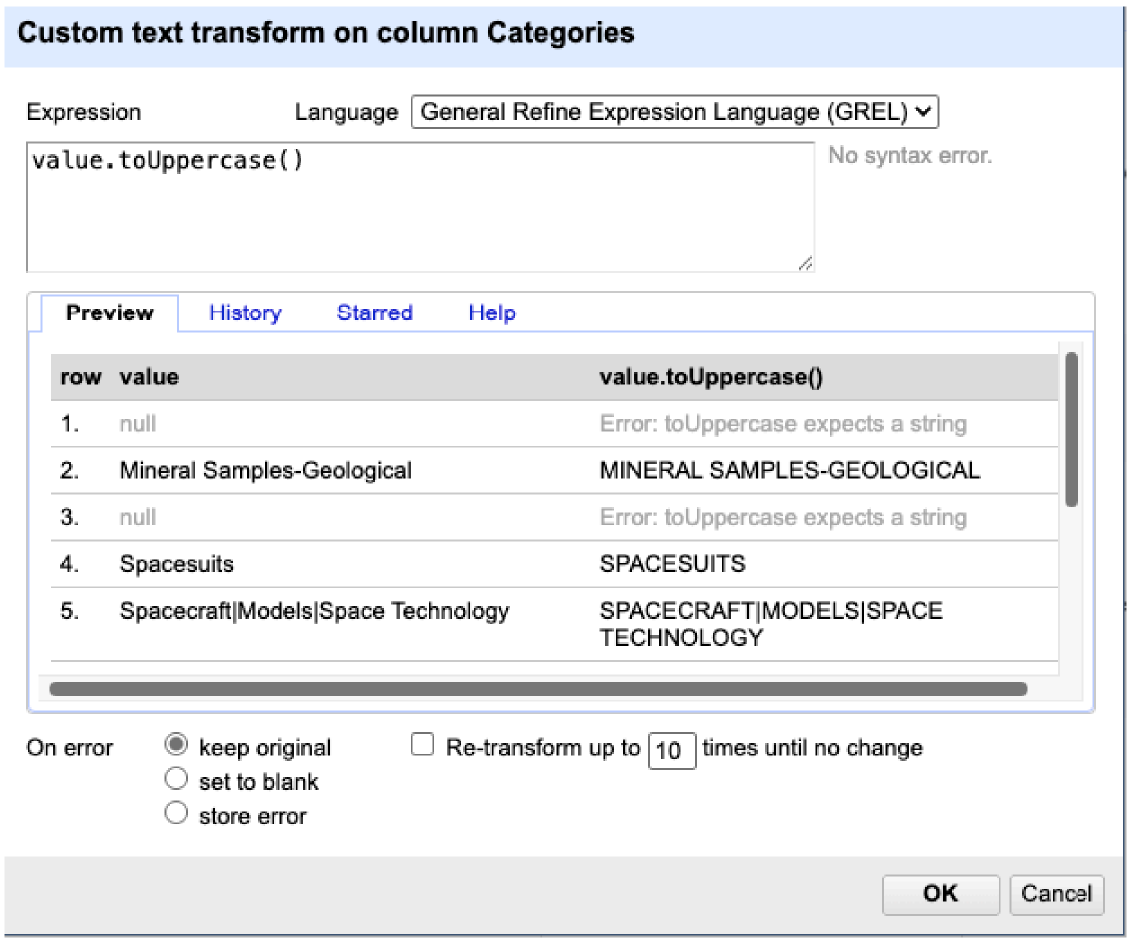
On error ● keep original ○ set to blank ○ store error

□ Re-transform up to 10 times until no change

Tasks –Part IV

1.Import data from the "phm-collection-4000.tsv" file into your OpenRefineproject

2.In the column “Categories” change values into uppercase



Custom text transform on column Categories

Expression Language: General Refine Expression Language (GREL)

```
value.toUppercase()
```

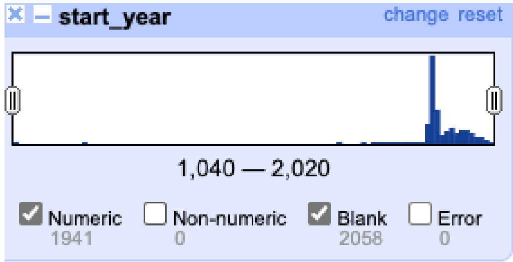
No syntax error.

row	value	value.toUppercase()
1.	null	Error: toUppercase expects a string
2.	Mineral Samples-Geological	MINERAL SAMPLES-GEOLOGICAL
3.	null	Error: toUppercase expects a string
4.	Spacesuits	SPACESUITS
5.	Spacecraft|Models|Space Technology	SPACECRAFT|MODELS|SPACE TECHNOLOGY

On error ● keep original ○ set to blank ○ store error

□ Re-transform up to 10 times until no change

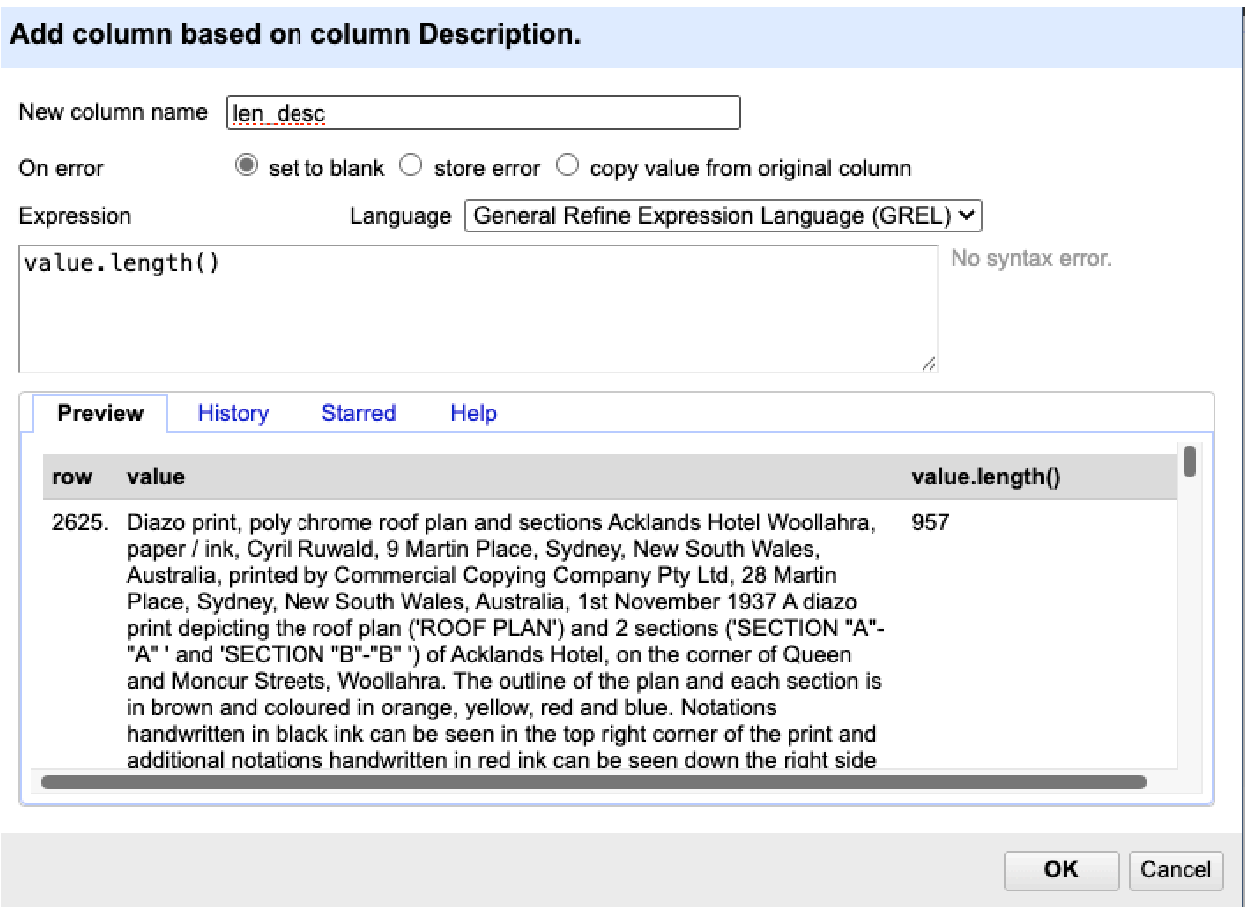
3.What is the oldest item in the dataset according to the provided information/data?



1,040 — 2,020

☑ Numeric 1941 □ Non-numeric 0 ☑ Blank 2058 □ Error 0

4.Create a new column, which contains the length of the "Description“ column



Add column based on column Description.

New column name: len_desc

On error: ● set to blank ○ store error ○ copy value from original column

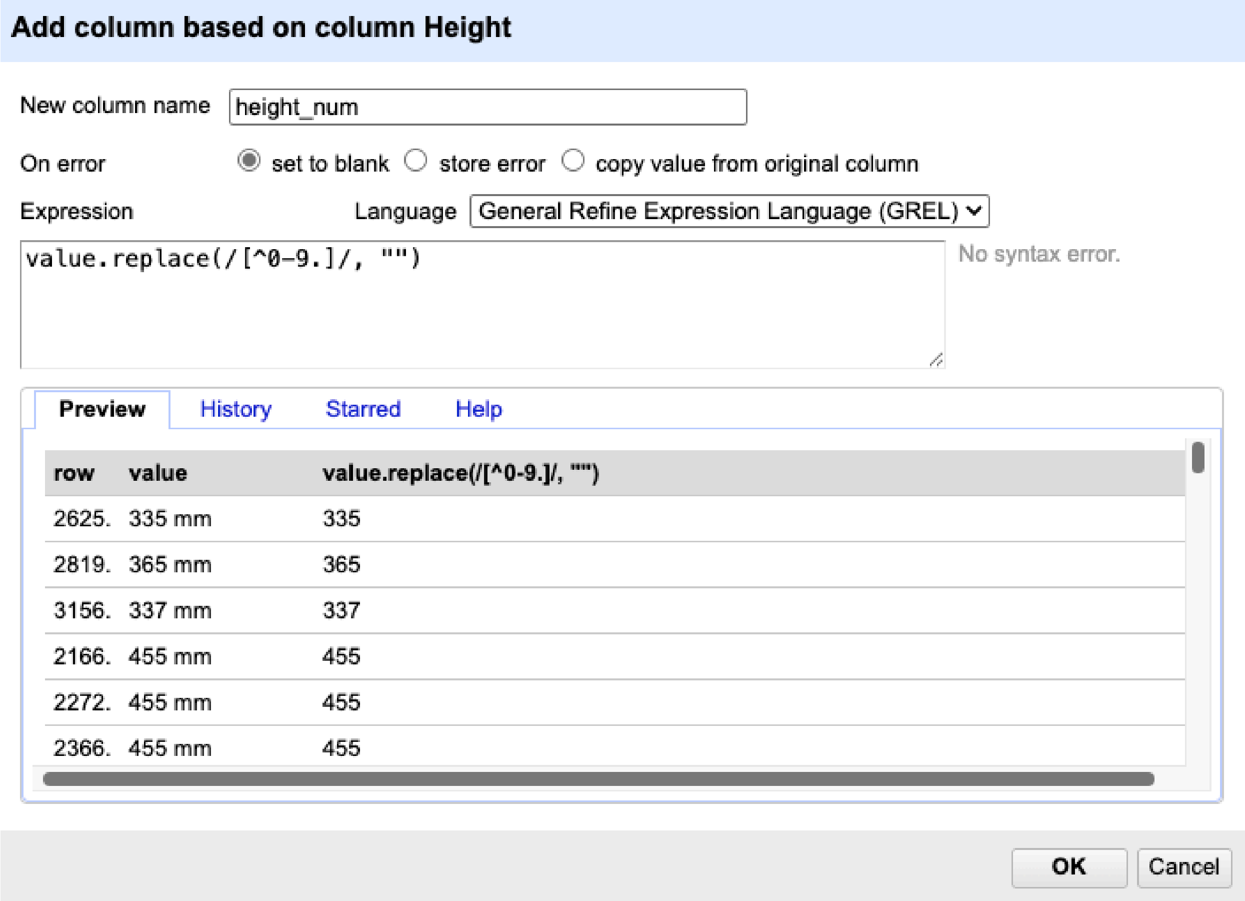
Expression — Language: General Refine Expression Language (GREL)

```
value.length()
```

No syntax error.

row	value	value.length()
2625.	Diazo print, poly chrome roof plan and sections Acklands Hotel Woollahra, paper / ink, Cyril Ruwald, 9 Martin Place, Sydney, New South Wales, Australia, printed by Commercial Copying Company Pty Ltd, 28 Martin Place, Sydney, New South Wales, Australia, 1st November 1937 A diazo print depicting the roof plan ('ROOF PLAN') and 2 sections ('SECTION "A"-"A" ' and 'SECTION "B"-"B" ') of Acklands Hotel, on the corner of Queen and Moncur Streets, Woollahra. The outline of the plan and each section is in brown and coloured in orange, yellow, red and blue. Notations handwritten in black ink can be seen in the top right corner of the print and additional notations handwritten in red ink can be seen down the right side	957

5.Create multiple new columns and keep only numbers inside newly created columns. Do it for the following columns "Height", "Width", "Depth"



Add column based on column Height

New column name: height_num

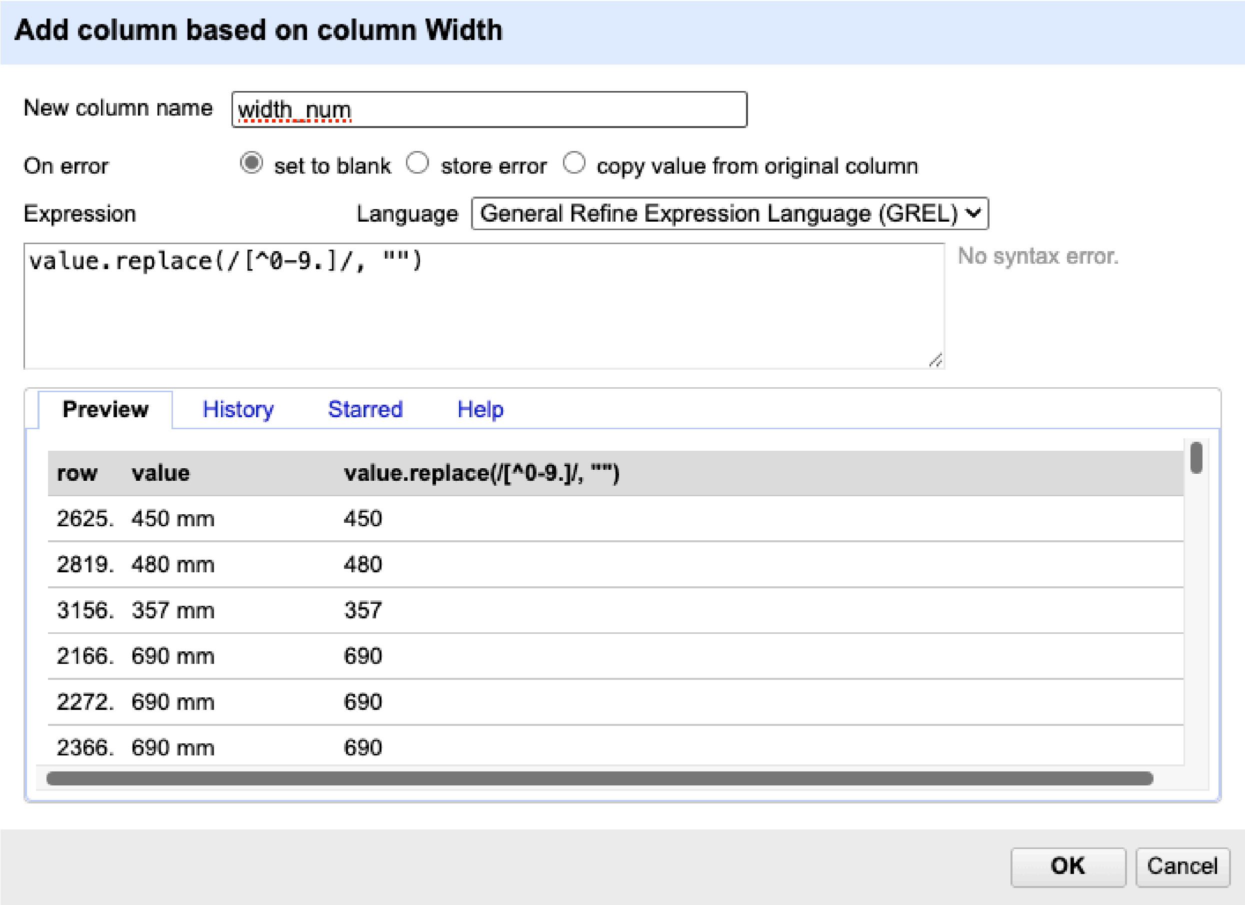
On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: General Refine Expression Language (GREL)

```
value.replace(/[^0-9.]/, "")
```

No syntax error.

row	value	value.replace(/[^0-9.]/, "")
2625.	335 mm	335
2819.	365 mm	365
3156.	337 mm	337
2166.	455 mm	455
2272.	455 mm	455
2366.	455 mm	455



Add column based on column Width

New column name: width_num

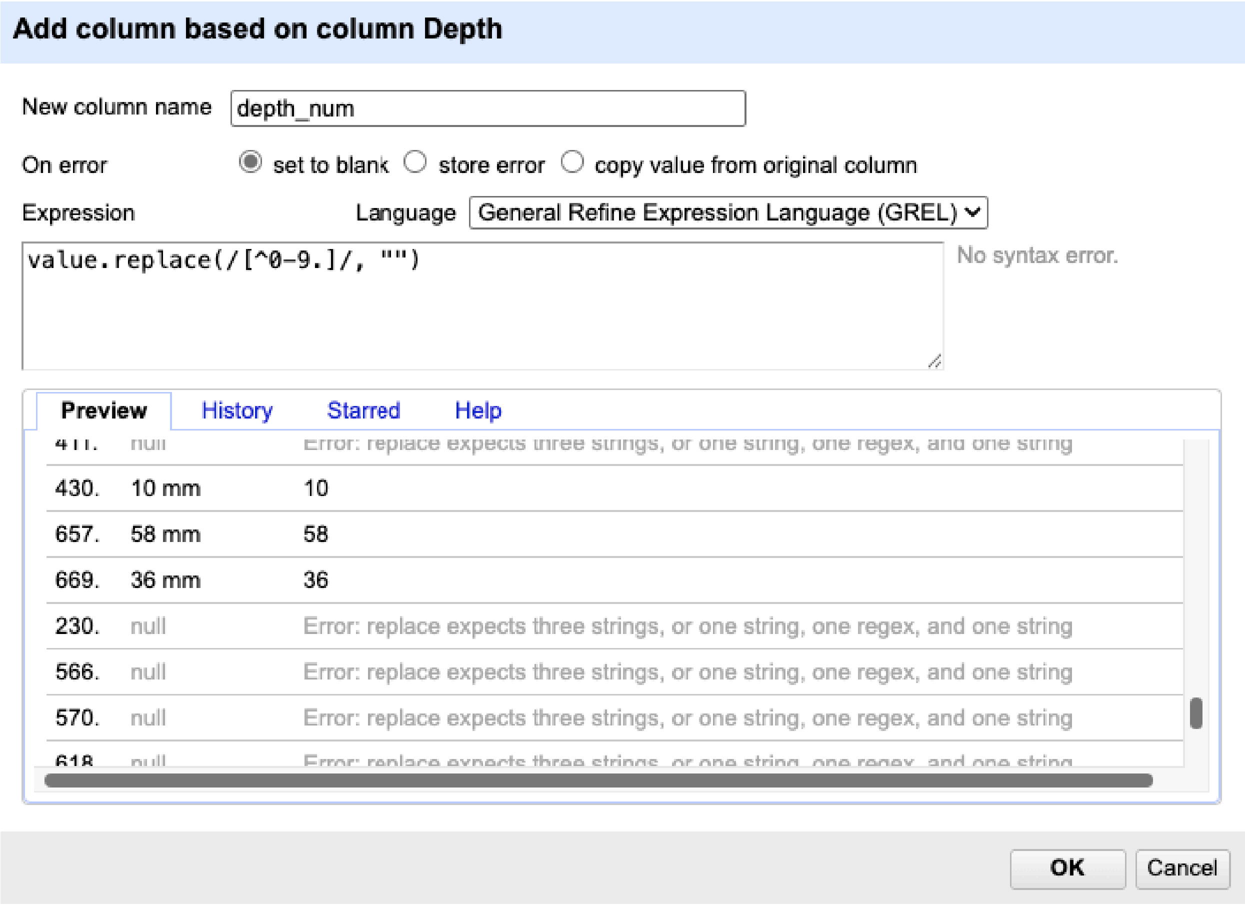
On error: ● set to blank ○ store error ○ copy value from original column

Expression Language: General Refine Expression Language (GREL)

```
value.replace(/[^0-9.]/, "")
```

No syntax error.

row	value	value.replace(/[^0-9.]/, "")
2625.	450 mm	450
2819.	480 mm	480
3156.	357 mm	357
2166.	690 mm	690
2272.	690 mm	690
2366.	690 mm	690



Add column based on column Depth

New column name: depth_num

On error: ● set to blank ○ store error ○ copy value from original column

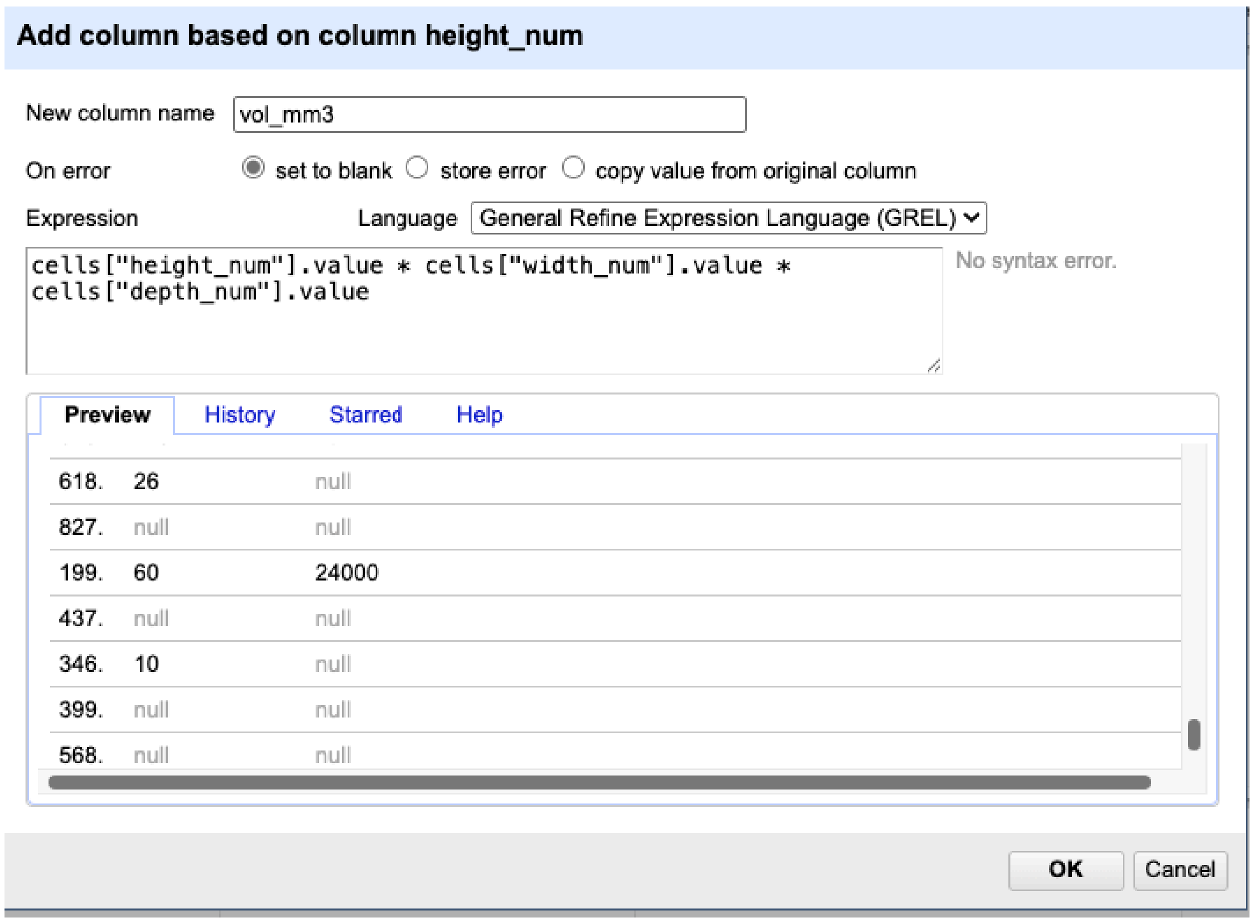
Expression Language: General Refine Expression Language (GREL)

```
value.replace(/[^0-9.]/, "")
```

No syntax error.

411.	null	Error: replace expects three strings, or one string, one regex, and one string
430.	10 mm	10
657.	58 mm	58
669.	36 mm	36
230.	null	Error: replace expects three strings, or one string, one regex, and one string
566.	null	Error: replace expects three strings, or one string, one regex, and one string
570.	null	Error: replace expects three strings, or one string, one regex, and one string
618.	null	Error: replace expects three strings, or one string, one regex, and one string

6.Create a new column that will calculate volume of each item (if measurements are provided). Which measurement unit should be used here?



Add column based on column height_num

New column name vol_mm3

On error ● set to blank ○ store error ○ copy value from original column

Expression Language General Refine Expression Language (GREL)

```
cells["height_num"].value * cells["width_num"].value *
cells["depth_num"].value
```

No syntax error.

618.	26	null
827.	null	null
199.	60	24000
437.	null	null
346.	10	null
399.	null	null
568.	null	null