

```
In [ ]: # Import dependencies
import pandas as pd
import numpy as np
pd.set_option('max_colwidth', 400)
```

Extract the crowdfunding.xlsx Data -Yiran Zhang

```
In [ ]: # Read the data into a Pandas DataFrame
crowdfunding_info_df = pd.read_excel('Resources/crowdfunding.xlsx')
crowdfunding_info_df.head()
```

Out[]:

| | cf_id | contact_id | company_name | blurb | goal | pledged | outcome | backers_count | country | currency | launched_at | deadline | staff_pick | spotlight | category & sub-category |
|---|-------|------------|-----------------------------|---|--------|---------|------------|---------------|---------|----------|-------------|------------|------------|-----------|-------------------------|
| 0 | 147 | 4661 | Baldwin, Riley and Jackson | Pre-emptive tertiary standardization | 100 | 0 | failed | 0 | CA | CAD | 1581573600 | 1614578400 | False | False | food/food trucks |
| 1 | 1621 | 3765 | Odom Inc | Managed bottom-line architecture | 1400 | 14560 | successful | 158 | US | USD | 1611554400 | 1621918800 | False | True | music/rock |
| 2 | 1812 | 4187 | Melton, Robinson and Fritz | Function-based leadingedge pricing structure | 108400 | 142523 | successful | 1425 | AU | AUD | 1608184800 | 1640844000 | False | False | technology/web |
| 3 | 2156 | 4941 | Mcdonald, Gonzalez and Ross | Vision-oriented fresh-thinking conglomeration | 4200 | 2477 | failed | 24 | US | USD | 1634792400 | 1642399200 | False | False | music/rock |
| 4 | 1365 | 2199 | Larson-Little | Proactive foreground core | 7600 | 5265 | failed | 53 | US | USD | 1608530400 | 1629694800 | False | False | theater/plays |

```
In [ ]: # Get a brief summary of the crowdfunding_info DataFrame.
crowdfunding_info_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   cf_id                  1000 non-null  int64
1   contact_id             1000 non-null  int64
2   company_name           1000 non-null  object
3   blurb                  1000 non-null  object
4   goal                   1000 non-null  int64
5   pledged                1000 non-null  int64
6   outcome                1000 non-null  object
7   backers_count          1000 non-null  int64
8   country                1000 non-null  object
9   currency               1000 non-null  object
10  launched_at            1000 non-null  int64
11  deadline               1000 non-null  int64
12  staff_pick             1000 non-null  bool
13  spotlight              1000 non-null  bool
14  category & sub-category 1000 non-null  object
dtypes: bool(2), int64(7), object(6)
memory usage: 103.6+ KB
```

Create the Category and Subcategory DataFrames

Create a Category DataFrame that has the following columns:

- A "category_id" column that is numbered sequential form 1 to the length of the number of unique categories.
- A "category" column that has only the categories.

Export the DataFrame as a `category.csv` CSV file.

Create a SubCategory DataFrame that has the following columns:

- A "subcategory_id" column that is numbered sequential form 1 to the length of the number of unique subcategories.
- A "subcategory" column that has only the subcategories.

Export the DataFrame as a `subcategory.csv` CSV file.

```
In [ ]: # Get the crowdfunding_info_df columns.
crowdfunding_info_df.columns

Out[ ]: Index(['cf_id', 'contact_id', 'company_name', 'blurb', 'goal', 'pledged',
            'outcome', 'backers_count', 'country', 'currency', 'launched_at',
            'deadline', 'staff_pick', 'spotlight', 'category & sub-category'],
            dtype='object')
```

```
In [ ]: # Assign the category and subcategory values to category and subcategory columns.
crowdfunding_info_df[['category', 'subcategory']] = crowdfunding_info_df['category & sub-category'].str.split('/', expand=True)
crowdfunding_info_df.head()
```

```
Out[ ]:
```

| | cf_id | contact_id | company_name | blurb | goal | pledged | outcome | backers_count | country | currency | launched_at | deadline | staff_pick | spotlight | category & sub-category | category | subcategory |
|---|-------|------------|-----------------------------|---|--------|---------|------------|---------------|---------|----------|-------------|------------|------------|-----------|-------------------------|------------|-------------|
| 0 | 147 | 4661 | Baldwin, Riley and Jackson | Pre-emptive tertiary standardization | 100 | 0 | failed | 0 | CA | CAD | 1581573600 | 1614578400 | False | False | food/food trucks | food | food trucks |
| 1 | 1621 | 3765 | Odom Inc | Managed bottom-line architecture | 1400 | 14560 | successful | 158 | US | USD | 1611554400 | 1621918800 | False | True | music/rock | music | rock |
| 2 | 1812 | 4187 | Melton, Robinson and Fritz | Function-based leadingedge pricing structure | 108400 | 142523 | successful | 1425 | AU | AUD | 1608184800 | 1640844000 | False | False | technology/web | technology | web |
| 3 | 2156 | 4941 | Mcdonald, Gonzalez and Ross | Vision-oriented fresh-thinking conglomeration | 4200 | 2477 | failed | 24 | US | USD | 1634792400 | 1642399200 | False | False | music/rock | music | rock |
| 4 | 1365 | 2199 | Larson-Little | Proactive foreground core | 7600 | 5265 | failed | 53 | US | USD | 1608530400 | 1629694800 | False | False | theater/plays | theater | plays |

```
In [ ]: # Get the unique categories and subcategories in separate lists.
categories = crowdfunding_info_df['category'].unique().tolist()
subcategories = crowdfunding_info_df['subcategory'].unique().tolist()

print(categories)
print(subcategories)

['food', 'music', 'technology', 'theater', 'film & video', 'publishing', 'games', 'photography', 'journalism']
['food trucks', 'rock', 'web', 'plays', 'documentary', 'electric music', 'drama', 'indie rock', 'wearables', 'nonfiction', 'animation', 'video games', 'shorts', 'fiction', 'photography books', 'radio & podcasts', 'metal', 'jazz', 'translations', 'television', 'mobile games', 'world music', 'science fiction', 'audio']

In [ ]: # Get the number of distinct values in the categories and subcategories lists.
print(len(categories))
print(len(subcategories))

9
24

In [ ]: # Create numpy arrays from 1-9 for the categories and 1-24 for the subcategories.
category_ids = np.arange(1, 10)
subcategory_ids = np.arange(1, 25)

print(category_ids)
print(subcategory_ids)

[1 2 3 4 5 6 7 8 9]
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]

In [ ]: # Use a list comprehension to add "cat" to each category_id.
cat_ids = ['cat' + str(id) for id in category_ids]

# Use a list comprehension to add "subcat" to each subcategory_id.
scat_ids = ['subcat' + str(id) for id in subcategory_ids]

print(cat_ids)
print(scat_ids)
```

```
['cat1', 'cat2', 'cat3', 'cat4', 'cat5', 'cat6', 'cat7', 'cat8', 'cat9']  
['subcat1', 'subcat2', 'subcat3', 'subcat4', 'subcat5', 'subcat6', 'subcat7', 'subcat8', 'subcat9', 'subcat10', 'subcat11', 'subcat12', 'subcat13', 'subcat14', 'subcat15', 'subcat16', 'subcat17', 'subcat18',  
'subcat19', 'subcat20', 'subcat21', 'subcat22', 'subcat23', 'subcat24']
```

```
In [ ]: # Create a category DataFrame with the category_id array as the category_id and categories List as the category name.  
category_df = pd.DataFrame({"category_id": cat_ids, "category": categories})  
  
# Create a category DataFrame with the subcategory_id array as the subcategory_id and subcategories List as the subcategory name.  
subcategory_df = pd.DataFrame({"subcategory_id": scat_ids, "subcategory": subcategories})
```

```
In [ ]: category_df
```

```
Out[ ]:
```

| | category_id | category |
|---|-------------|--------------|
| 0 | cat1 | food |
| 1 | cat2 | music |
| 2 | cat3 | technology |
| 3 | cat4 | theater |
| 4 | cat5 | film & video |
| 5 | cat6 | publishing |
| 6 | cat7 | games |
| 7 | cat8 | photography |
| 8 | cat9 | journalism |

```
In [ ]: subcategory_df
```

```
Out[ ]:
```

| | subcategory_id | subcategory |
|----|----------------|-------------------|
| 0 | subcat1 | food trucks |
| 1 | subcat2 | rock |
| 2 | subcat3 | web |
| 3 | subcat4 | plays |
| 4 | subcat5 | documentary |
| 5 | subcat6 | electric music |
| 6 | subcat7 | drama |
| 7 | subcat8 | indie rock |
| 8 | subcat9 | wearables |
| 9 | subcat10 | nonfiction |
| 10 | subcat11 | animation |
| 11 | subcat12 | video games |
| 12 | subcat13 | shorts |
| 13 | subcat14 | fiction |
| 14 | subcat15 | photography books |
| 15 | subcat16 | radio & podcasts |
| 16 | subcat17 | metal |
| 17 | subcat18 | jazz |
| 18 | subcat19 | translations |
| 19 | subcat20 | television |
| 20 | subcat21 | mobile games |
| 21 | subcat22 | world music |
| 22 | subcat23 | science fiction |
| 23 | subcat24 | audio |

```
In [ ]: # Export categories_df and subcategories_df as CSV files.
category_df.to_csv("Resources/category.csv", index=False)

subcategory_df.to_csv("Resources/subcategory.csv", index=False)
```

Campaign DataFrame -Sean Daily

Create a Campaign DataFrame that has the following columns:

- The "cf_id" column.
- The "contact_id" column.
- The "company_name" column.
- The "blurb" column is renamed as "description."
- The "goal" column.
- The "goal" column is converted to a float datatype.
- The "pledged" column is converted to a float datatype.
- The "backers_count" column.
- The "country" column.
- The "currency" column.

- The "launched_at" column is renamed as "launch_date" and converted to a datetime format.
- The "deadline" column is renamed as "end_date" and converted to a datetime format.
- The "category_id" with the unique number matching the "category_id" from the category DataFrame.
- The "subcategory_id" with the unique number matching the "subcategory_id" from the subcategory DataFrame.
- And, create a column that contains the unique four-digit contact ID number from the `contact.xlsx` file.

Then export the DataFrame as a `campaign.csv` CSV file.

```
In [ ]: # Create a copy of the crowdfunding_info_df DataFrame name campaign_df.
campaign_df = crowdfunding_info_df.copy()
campaign_df.head()
```

```
Out[ ]:
```

| | cf_id | contact_id | company_name | blurb | goal | pledged | outcome | backers_count | country | currency | launched_at | deadline | staff_pick | spotlight | category & sub-category | category | subcategory |
|---|-------|------------|-----------------------------|---|--------|---------|------------|---------------|---------|----------|-------------|------------|------------|-----------|-------------------------|------------|-------------|
| 0 | 147 | 4661 | Baldwin, Riley and Jackson | Pre-emptive tertiary standardization | 100 | 0 | failed | 0 | CA | CAD | 1581573600 | 1614578400 | False | False | food/food trucks | food | food trucks |
| 1 | 1621 | 3765 | Odom Inc | Managed bottom-line architecture | 1400 | 14560 | successful | 158 | US | USD | 1611554400 | 1621918800 | False | True | music/rock | music | rock |
| 2 | 1812 | 4187 | Melton, Robinson and Fritz | Function-based leadingedge pricing structure | 108400 | 142523 | successful | 1425 | AU | AUD | 1608184800 | 1640844000 | False | False | technology/web | technology | web |
| 3 | 2156 | 4941 | Mcdonald, Gonzalez and Ross | Vision-oriented fresh-thinking conglomeration | 4200 | 2477 | failed | 24 | US | USD | 1634792400 | 1642399200 | False | False | music/rock | music | rock |
| 4 | 1365 | 2199 | Larson-Little | Proactive foreground core | 7600 | 5265 | failed | 53 | US | USD | 1608530400 | 1629694800 | False | False | theater/plays | theater | plays |

```
In [ ]: # Rename the blurb, launched_at, and deadline columns.
campaign_df = campaign_df.rename(columns={'blurb': 'description', 'launched_at': 'launch_date', 'deadline': 'end_date'})
campaign_df
```

Out[]:

| | cf_id | contact_id | company_name | description | goal | pledged | outcome | backers_count | country | currency | launch_date | end_date | staff_pick | spotlight | category & sub-category | category | subcategory |
|-----|-------|------------|------------------------------|---|--------|---------|------------|---------------|---------|----------|-------------|------------|------------|-----------|-------------------------|------------|-------------|
| 0 | 147 | 4661 | Baldwin, Riley and Jackson | Pre-emptive tertiary standardization | 100 | 0 | failed | 0 | CA | CAD | 1581573600 | 1614578400 | False | False | food/food trucks | food | food trucks |
| 1 | 1621 | 3765 | Odom Inc | Managed bottom-line architecture | 1400 | 14560 | successful | 158 | US | USD | 1611554400 | 1621918800 | False | True | music/rock | music | rock |
| 2 | 1812 | 4187 | Melton, Robinson and Fritz | Function-based leadingedge pricing structure | 108400 | 142523 | successful | 1425 | AU | AUD | 1608184800 | 1640844000 | False | False | technology/web | technology | web |
| 3 | 2156 | 4941 | Mcdonald, Gonzalez and Ross | Vision-oriented fresh-thinking conglomeration | 4200 | 2477 | failed | 24 | US | USD | 1634792400 | 1642399200 | False | False | music/rock | music | rock |
| 4 | 1365 | 2199 | Larson-Little | Proactive foreground core | 7600 | 5265 | failed | 53 | US | USD | 1608530400 | 1629694800 | False | False | theater/plays | theater | plays |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 995 | 2986 | 3684 | Manning-Hamilton | Vision-oriented scalable definition | 97300 | 153216 | successful | 2043 | US | USD | 1609221600 | 1622350800 | False | True | food/food trucks | food | food trucks |
| 996 | 2031 | 5784 | Butler LLC | Future-proofed upward-trending migration | 6600 | 4814 | failed | 112 | US | USD | 1634274000 | 1638252000 | False | False | theater/plays | theater | plays |
| 997 | 1627 | 1498 | Ball LLC | Right-sized full-range throughput | 7600 | 4603 | canceled | 139 | IT | EUR | 1636174800 | 1639116000 | False | False | theater/plays | theater | plays |
| 998 | 2175 | 6073 | Taylor, Santiago and Flores | Polarized composite customer loyalty | 66600 | 37823 | failed | 374 | US | USD | 1602133200 | 1618117200 | False | True | music/indie rock | music | indie rock |
| 999 | 1788 | 4939 | Hernandez, Norton and Kelley | Expanded eco-centric policy | 111100 | 62819 | canceled | 1122 | US | USD | 1609308000 | 1629262800 | False | False | food/food trucks | food | food trucks |

1000 rows × 17 columns

In []:

```
# Convert the goal and pledged columns to a `float` data type.
campaign_df['goal'] = campaign_df['goal'].astype(float)
campaign_df['pledged'] = campaign_df['pledged'].astype(float)
campaign_df
```

Out[]:

| | cf_id | contact_id | company_name | description | goal | pledged | outcome | backers_count | country | currency | launch_date | end_date | staff_pick | spotlight | category & sub-category | category | subcategory |
|-----|-------|------------|------------------------------|---|----------|----------|------------|---------------|---------|----------|-------------|------------|------------|-----------|-------------------------|------------|-------------|
| 0 | 147 | 4661 | Baldwin, Riley and Jackson | Pre-emptive tertiary standardization | 100.0 | 0.0 | failed | 0 | CA | CAD | 1581573600 | 1614578400 | False | False | food/food trucks | food | food trucks |
| 1 | 1621 | 3765 | Odom Inc | Managed bottom-line architecture | 1400.0 | 14560.0 | successful | 158 | US | USD | 1611554400 | 1621918800 | False | True | music/rock | music | rock |
| 2 | 1812 | 4187 | Melton, Robinson and Fritz | Function-based leadingedge pricing structure | 108400.0 | 142523.0 | successful | 1425 | AU | AUD | 1608184800 | 1640844000 | False | False | technology/web | technology | web |
| 3 | 2156 | 4941 | Mcdonald, Gonzalez and Ross | Vision-oriented fresh-thinking conglomeration | 4200.0 | 2477.0 | failed | 24 | US | USD | 1634792400 | 1642399200 | False | False | music/rock | music | rock |
| 4 | 1365 | 2199 | Larson-Little | Proactive foreground core | 7600.0 | 5265.0 | failed | 53 | US | USD | 1608530400 | 1629694800 | False | False | theater/plays | theater | plays |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 995 | 2986 | 3684 | Manning-Hamilton | Vision-oriented scalable definition | 97300.0 | 153216.0 | successful | 2043 | US | USD | 1609221600 | 1622350800 | False | True | food/food trucks | food | food trucks |
| 996 | 2031 | 5784 | Butler LLC | Future-proofed upward-trending migration | 6600.0 | 4814.0 | failed | 112 | US | USD | 1634274000 | 1638252000 | False | False | theater/plays | theater | plays |
| 997 | 1627 | 1498 | Ball LLC | Right-sized full-range throughput | 7600.0 | 4603.0 | canceled | 139 | IT | EUR | 1636174800 | 1639116000 | False | False | theater/plays | theater | plays |
| 998 | 2175 | 6073 | Taylor, Santiago and Flores | Polarized composite customer loyalty | 66600.0 | 37823.0 | failed | 374 | US | USD | 1602133200 | 1618117200 | False | True | music/indie rock | music | indie rock |
| 999 | 1788 | 4939 | Hernandez, Norton and Kelley | Expanded eco-centric policy | 111100.0 | 62819.0 | canceled | 1122 | US | USD | 1609308000 | 1629262800 | False | False | food/food trucks | food | food trucks |

1000 rows × 17 columns

In []:

```
# Check the datatypes
campaign_df.dtypes
```

Out[]:

| | |
|-------------------------|---------|
| cf_id | int64 |
| contact_id | int64 |
| company_name | object |
| description | object |
| goal | float64 |
| pledged | float64 |
| outcome | object |
| backers_count | int64 |
| country | object |
| currency | object |
| launch_date | int64 |
| end_date | int64 |
| staff_pick | bool |
| spotlight | bool |
| category & sub-category | object |
| category | object |
| subcategory | object |
| dtype: | object |

In []:

```
# Format the launched_date and end_date columns to datetime format
from datetime import datetime
campaign_df['launch_date'] = pd.to_datetime(campaign_df['launch_date'], unit='s', errors='coerce')
campaign_df['end_date'] = pd.to_datetime(campaign_df['end_date'], unit='s', errors='coerce')
campaign_df
```

Out[]:

| | cf_id | contact_id | company_name | description | goal | pledged | outcome | backers_count | country | currency | launch_date | end_date | staff_pick | spotlight | category & sub-category | category | subcategory |
|-----|-------|------------|------------------------------|---|----------|----------|------------|---------------|---------|----------|---------------------|---------------------|------------|-----------|-------------------------|------------|-------------|
| 0 | 147 | 4661 | Baldwin, Riley and Jackson | Pre-emptive tertiary standardization | 100.0 | 0.0 | failed | 0 | CA | CAD | 2020-02-13 06:00:00 | 2021-03-01 06:00:00 | False | False | food/food trucks | food | food trucks |
| 1 | 1621 | 3765 | Odom Inc | Managed bottom-line architecture | 1400.0 | 14560.0 | successful | 158 | US | USD | 2021-01-25 06:00:00 | 2021-05-25 05:00:00 | False | True | music/rock | music | rock |
| 2 | 1812 | 4187 | Melton, Robinson and Fritz | Function-based leadingedge pricing structure | 108400.0 | 142523.0 | successful | 1425 | AU | AUD | 2020-12-17 06:00:00 | 2021-12-30 06:00:00 | False | False | technology/web | technology | web |
| 3 | 2156 | 4941 | Mcdonald, Gonzalez and Ross | Vision-oriented fresh-thinking conglomeration | 4200.0 | 2477.0 | failed | 24 | US | USD | 2021-10-21 05:00:00 | 2022-01-17 06:00:00 | False | False | music/rock | music | rock |
| 4 | 1365 | 2199 | Larson-Little | Proactive foreground core | 7600.0 | 5265.0 | failed | 53 | US | USD | 2020-12-21 06:00:00 | 2021-08-23 05:00:00 | False | False | theater/plays | theater | plays |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 995 | 2986 | 3684 | Manning-Hamilton | Vision-oriented scalable definition | 97300.0 | 153216.0 | successful | 2043 | US | USD | 2020-12-29 06:00:00 | 2021-05-30 05:00:00 | False | True | food/food trucks | food | food trucks |
| 996 | 2031 | 5784 | Butler LLC | Future-proofed upward-trending migration | 6600.0 | 4814.0 | failed | 112 | US | USD | 2021-10-15 05:00:00 | 2021-11-30 06:00:00 | False | False | theater/plays | theater | plays |
| 997 | 1627 | 1498 | Ball LLC | Right-sized full-range throughput | 7600.0 | 4603.0 | canceled | 139 | IT | EUR | 2021-11-06 05:00:00 | 2021-12-10 06:00:00 | False | False | theater/plays | theater | plays |
| 998 | 2175 | 6073 | Taylor, Santiago and Flores | Polarized composite customer loyalty | 66600.0 | 37823.0 | failed | 374 | US | USD | 2020-10-08 05:00:00 | 2021-04-11 05:00:00 | False | True | music/indie rock | music | indie rock |
| 999 | 1788 | 4939 | Hernandez, Norton and Kelley | Expanded eco-centric policy | 111100.0 | 62819.0 | canceled | 1122 | US | USD | 2020-12-30 06:00:00 | 2021-08-18 05:00:00 | False | False | food/food trucks | food | food trucks |

1000 rows × 17 columns

In []:

```
# Merge the campaign_df with the category_df on the "category" column and
# the subcategory_df on the "subcategory" column.
campaign_merged_df = pd.merge(campaign_df, category_df, how='left', on='category')
final_campaign_merged_df = pd.merge(campaign_merged_df, subcategory_df, how='left', on='subcategory')
final_campaign_merged_df.tail(10)
```


Out []:

| | cf_id | contact_id | company_name | description | goal | pledged | outcome | backers_count | country | currency | launch_date | end_date | staff_pick | spotlight | category & sub-category | category | subcategory | category_ |
|-----|-------|------------|------------------------------|--|----------|----------|------------|---------------|---------|----------|---------------------|---------------------|------------|-----------|-------------------------------|--------------|-------------------|-----------|
| 990 | 1561 | 5500 | Ortiz-Roberts | Devolved foreground customer loyalty | 7800.0 | 6839.0 | failed | 64 | US | USD | 2021-06-09 05:00:00 | 2021-06-18 05:00:00 | False | True | film & video/drama | film & video | drama | ca |
| 991 | 2632 | 3494 | Ramirez LLC | Reduced reciprocal focus group | 9800.0 | 11091.0 | successful | 241 | US | USD | 2020-12-09 06:00:00 | 2021-05-26 05:00:00 | False | True | music/rock | music | rock | ca |
| 992 | 439 | 3924 | Morrow Inc | Networked global migration | 3100.0 | 13223.0 | successful | 132 | US | USD | 2020-06-14 05:00:00 | 2021-02-09 06:00:00 | False | True | film & video/drama | film & video | drama | ca |
| 993 | 461 | 3521 | Erickson-Rogers | De-engineered even-keeled definition | 9800.0 | 7608.0 | canceled | 75 | IT | EUR | 2021-07-03 05:00:00 | 2021-07-08 05:00:00 | False | True | photography/photography books | photography | photography books | ca |
| 994 | 1419 | 5262 | Leach, Rich and Price | Implemented bi-directional flexibility | 141100.0 | 74073.0 | failed | 842 | US | USD | 2021-11-15 06:00:00 | 2021-12-07 06:00:00 | False | True | publishing/translations | publishing | translations | ca |
| 995 | 2986 | 3684 | Manning-Hamilton | Vision-oriented scalable definition | 97300.0 | 153216.0 | successful | 2043 | US | USD | 2020-12-29 06:00:00 | 2021-05-30 05:00:00 | False | True | food/food trucks | food | food trucks | ca |
| 996 | 2031 | 5784 | Butler LLC | Future-proofed upward-trending migration | 6600.0 | 4814.0 | failed | 112 | US | USD | 2021-10-15 05:00:00 | 2021-11-30 06:00:00 | False | False | theater/plays | theater | plays | ca |
| 997 | 1627 | 1498 | Ball LLC | Right-sized full-range throughput | 7600.0 | 4603.0 | canceled | 139 | IT | EUR | 2021-11-06 05:00:00 | 2021-12-10 06:00:00 | False | False | theater/plays | theater | plays | ca |
| 998 | 2175 | 6073 | Taylor, Santiago and Flores | Polarized composite customer loyalty | 66600.0 | 37823.0 | failed | 374 | US | USD | 2020-10-08 05:00:00 | 2021-04-11 05:00:00 | False | True | music/indie rock | music | indie rock | ca |
| 999 | 1788 | 4939 | Hernandez, Norton and Kelley | Expanded eco-centric policy | 111100.0 | 62819.0 | canceled | 1122 | US | USD | 2020-12-30 06:00:00 | 2021-08-18 05:00:00 | False | False | food/food trucks | food | food trucks | ca |

In []:

```
# Drop unwanted columns
clean_campaign_df = final_campaign_merged_df.drop(["staff_pick", "spotlight", "category & sub-category", "category", "subcategory"], axis = 1)
clean_campaign_df
```

Out []:

| | cf_id | contact_id | company_name | | description | goal | pledged | outcome | backers_count | country | currency | launch_date | end_date | category_id | subcategory_id |
|-----|-------|------------|------------------------------|--|---|----------|----------|------------|---------------|---------|----------|---------------------|---------------------|-------------|----------------|
| 0 | 147 | 4661 | Baldwin, Riley and Jackson | | Pre-emptive tertiary standardization | 100.0 | 0.0 | failed | 0 | CA | CAD | 2020-02-13 06:00:00 | 2021-03-01 06:00:00 | cat1 | subcat1 |
| 1 | 1621 | 3765 | Odom Inc | | Managed bottom-line architecture | 1400.0 | 14560.0 | successful | 158 | US | USD | 2021-01-25 06:00:00 | 2021-05-25 05:00:00 | cat2 | subcat2 |
| 2 | 1812 | 4187 | Melton, Robinson and Fritz | | Function-based leadingedge pricing structure | 108400.0 | 142523.0 | successful | 1425 | AU | AUD | 2020-12-17 06:00:00 | 2021-12-30 06:00:00 | cat3 | subcat3 |
| 3 | 2156 | 4941 | Mcdonald, Gonzalez and Ross | | Vision-oriented fresh-thinking conglomeration | 4200.0 | 2477.0 | failed | 24 | US | USD | 2021-10-21 05:00:00 | 2022-01-17 06:00:00 | cat2 | subcat2 |
| 4 | 1365 | 2199 | Larson-Little | | Proactive foreground core | 7600.0 | 5265.0 | failed | 53 | US | USD | 2020-12-21 06:00:00 | 2021-08-23 05:00:00 | cat4 | subcat4 |
| ... | ... | ... | ... | | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 995 | 2986 | 3684 | Manning-Hamilton | | Vision-oriented scalable definition | 97300.0 | 153216.0 | successful | 2043 | US | USD | 2020-12-29 06:00:00 | 2021-05-30 05:00:00 | cat1 | subcat1 |
| 996 | 2031 | 5784 | Butler LLC | | Future-proofed upward-trending migration | 6600.0 | 4814.0 | failed | 112 | US | USD | 2021-10-15 05:00:00 | 2021-11-30 06:00:00 | cat4 | subcat4 |
| 997 | 1627 | 1498 | Ball LLC | | Right-sized full-range throughput | 7600.0 | 4603.0 | canceled | 139 | IT | EUR | 2021-11-06 05:00:00 | 2021-12-10 06:00:00 | cat4 | subcat4 |
| 998 | 2175 | 6073 | Taylor, Santiago and Flores | | Polarized composite customer loyalty | 66600.0 | 37823.0 | failed | 374 | US | USD | 2020-10-08 05:00:00 | 2021-04-11 05:00:00 | cat2 | subcat8 |
| 999 | 1788 | 4939 | Hernandez, Norton and Kelley | | Expanded eco-centric policy | 111100.0 | 62819.0 | canceled | 1122 | US | USD | 2020-12-30 06:00:00 | 2021-08-18 05:00:00 | cat1 | subcat1 |

1000 rows × 14 columns

In []:

```
# Export the DataFrame as a CSV file.
clean_campaign_df.to_csv("Resources/campaign.csv", index=False)
```

Extract the contacts.xlsx Data. -Michelle Moody

In []:

```
# Read the data into a Pandas DataFrame. Use the `header=3` parameter when reading in the data; different header format from the crowdfunding file
contact_info_df_orig = pd.read_excel('Resources/contacts.xlsx', header=3)
contact_info_df_orig.head()
```

Out []:

| | contact_info |
|---|---|
| 0 | {"contact_id": 4661, "name": "Cecilia Velasco", "email": "cecilia.velasco@rodrigues.fr"} |
| 1 | {"contact_id": 3765, "name": "Mariana Ellis", "email": "mariana.ellis@rossi.org"} |
| 2 | {"contact_id": 4187, "name": "Sofie Woods", "email": "sofie.woods@riviere.com"} |
| 3 | {"contact_id": 4941, "name": "Jeanette Iannotti", "email": "jeanette.iannotti@yahoo.com"} |
| 4 | {"contact_id": 2199, "name": "Samuel Sorgatz", "email": "samuel.sorgatz@gmail.com"} |

In []:

```
# Get the column names and datatypes using `.info()` - One row = Dictionary format
contact_info_df_orig.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 1 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    contact_info 1000 non-null   object
dtypes: object(1)
memory usage: 7.9+ KB
```

Create the Contacts DataFrame

Create a **Contacts DataFrame** that has the following columns:

- A column named "contact_id" that contains the unique number of the contact person.
- A column named "first_name" that contains the first name of the contact person.
- A column named "last_name" that contains the first name of the contact person.
- A column named "email" that contains the email address of the contact person

Then export the DataFrame as a `contacts.csv` CSV file.

Option 1: Use Pandas to create the contacts DataFrame.

```
In [ ]: # Test extracting the key and values from the first row using json.loads().
import json
```

```
first_row = contact_info_df_orig.iloc[0]['contact_info']
converted_data = json.loads(first_row)
```

```
# Convert the data to a Python dictionary.
for k, v in converted_data.items():
    print(k, v)
```

```
contact_id 4661
name Cecilia Velasco
email cecilia.velasco@rodrigues.fr
```

```
In [ ]: # Iterate through the contact_info_df and convert each row to a dictionary.
dict_values = []
for i, row in contact_info_df_orig.iterrows():

    # Get the data first item in each row
    data = row['contact_info']
    converted_data = json.loads(data)

    # Iterate through each dictionary (row) and get the values for each row using List comprehension
    row_values = [v for k, v in converted_data.items()]

    # Append the list of values for each row to a new list
    dict_values.append(row_values)

# Print out the List of values for each row
print(dict_values)
```

[4661, 'Cecilia Velasco', 'cecilia.velasco@rodrigues.fr'], [3765, 'Mariana Ellis', 'mariana.ellis@rossi.org'], [4187, 'Sofie Woods', 'sofie.woods@riviere.com'], [4941, 'Jeanette Iannotti', 'jeanette.iannotti@yahoo.com'], [2199, 'Samuel Sorgatz', 'samuel.sorgatz@gmail.com'], [5650, 'Socorro Luna', 'socorro.luna@hotmail.com'], [5889, 'Carolina Murray', 'carolina.murray@knight.com'], [4842, 'Kayla Moon', 'kayla.moon@yahoo.de'], [3280, 'Ariadna Geisel', 'ariadna.geisel@rangel.com'], [5468, 'Danielle Ladeck', 'danielle.ladeck@scalparo.net'], [3064, 'Tatiana Thompson', 'tatiana.thompson@hunt.net'], [4904, 'Caleb Benavides', 'caleb.benavides@rubio.com'], [1299, 'Sandra Hardy', 'sandra.hardy@web.de'], [5602, 'Lotti Morris', 'lotti.morris@yahoo.co.uk'], [5753, 'Reinhilde White', 'reinhilde.white@voila.fr'], [4495, 'Kerry Patel', 'kerry.patel@hutchinson.com'], [4269, 'Sophie Antoine', 'sophie.antoine@andersen.com'], [2226, 'Martha Girard', 'martha.girard@web.de'], [1558, 'Stephanie King', 'stephanie.king@cervantes.com'], [2307, 'Amanda Palmer', 'amanda.palmer@didier.fr'], [2900, 'Lina Alcalá', 'lina.alcala@vespa.net'], [5695, 'Itzel Murphy', 'itzel.murphy@muellen.de'], [5708, 'Filippo Parry', 'filippo.parry@live.com'], [1663, 'Katelyn Cole', 'katelyn.cole@fielig.com'], [3605, 'Brian Novak', 'brian.novak@ford.net'], [4678, 'Cilly Gay', 'cilly.gay@callegaro.it'], [2251, 'Yolanda Snyder', 'yolanda.snyder@gmx.de'], [6202, 'Evelin Odone', 'evelin.odone@ibarra.net'], [3715, 'Ingeborg Alba', 'ingeborg.alba@hotmail.com'], [4242, 'Marina Madrid', 'marina.madrid@galarza-alba.com'], [4326, 'Sheila Goodwin', 'sheila.goodwin@yahoo.com'], [5560, 'Valeria Rich', 'valeria.rich@turchetta-mondadori.it'], [4002, 'Dustin Camacho', 'dustin.camacho@rhodes.org.au'], [3813, 'Amalia Marenzio', 'amalia.marenzio@grupo.com'], [5336, 'Gian Long', 'gian.long@hotmail.com'], [4994, 'Stewart Hunt', 'stewart.hunt@anderson-vargas.biz'], [1471, 'Greca Ruiz', 'greca.ruiz@carri.co.uk'], [4482, 'Gerald Olivera', 'gerald.olivera@outlook.com'], [3241, 'Jaqueline Wallace', 'jaqueline.wallace@gmail.com'], [3477, 'John Lane', 'john.lane@gregoire.fr'], [2265, 'Pero Joly', 'pero.joly@bernard.net'], [5911, 'Chad Turner', 'chad.turner@gmail.com'], [2288, 'Adam Zavala', 'adam.zavala@guicard.fr'], [4064, 'Tyler Rivera', 'tyler.rivera@guajardo-ozuna.com'], [1294, 'Jens Graham', 'jens.graham@jones-buckley.com'], [5008, 'Virginia Caetani', 'virginia.caetani@sosa.biz'], [3604, 'Martino Wagner', 'martino.wagner@laposte.net'], [3263, 'Martin Meyer', 'martin.meyer@davis.co.uk'], [5631, 'Marguerite Walls', 'marguerite.walls@martinez.fr'], [2851, 'Peter Vogt', 'peter.vogt@yahoo.com'], [3714, 'Nicholas Christian', 'nicholas.christian@hotmail.de'], [1664, 'Susi Steinberg', 'susi.steinberg@preiss.com'], [5027, 'Tammy Ramazzotti', 'tammy.ramazzotti@gmail.com'], [3070, 'Abdul Thomas', 'abdul.thomas@vasari.com'], [4248, 'Justin Luxardo', 'justin.luxardo@googletel.com'], [2034, 'Antonio Gibson', 'antonio.gibson@faust.net'], [4085, 'Jeremy Gomez', 'jeremy.gomez@oconnor.org'], [3569, 'Raymond Solorzano', 'raymond.solorzano@gmail.com'], [3889, 'Flora Harris', 'flora.harris@hotmail.com.au'], [3136, 'Cheryl Boyd', 'cheryl.boyd@segni.it'], [2103, 'Rhys Leiva', 'rhys.leiva@gmx.de'], [2329, 'Mariano Prieto', 'mariano.prieto@vodafone.it'], [3325, 'Laurie Fibonacci', 'laurie.fibonacci@gmail.com'], [3131, 'David Rudolph', 'david.rudolph@nelson.com'], [4995, 'Bryan Ross', 'bryan.ross@yahoo.com'], [3631, 'Ubaldo Brown', 'ubaldo.brown@philippe.com'], [5373, 'Clelia Faulkner', 'clelia.faulkner@hotmail.com'], [3126, 'Sara Erickson', 'sara.erickson@yahoo.de'], [2194, 'Rupert Valle', 'rupert.valle@figolamo.it'], [2906, 'Puccio Kitzmann', 'puccio.kitzmann@yahoo.com'], [2611, 'Thomas Hutchinson', 'thomas.hutchinson@gmx.de'], [2374, 'Abelardo Castro', 'abelardo.castro@gmail.com'], [3254, 'Greco Walker', 'greco.walker@yahoo.com'], [3571, 'Miranda Lacombe', 'miranda.lacombe@longoria-vanegas.net'], [2812, 'Diana Schuchardt', 'diana.schuchardt@libero.it'], [3961, 'Stacey Knox', 'stacey.knox@corporacin.com'], [3872, 'Glen Faivre', 'glen.faivre@yahoo.com'], [4736, 'Andrew Torres', 'andrew.torres@ruiz-torres.org'], [5119, 'Denny Fritz', 'denny.fritz@despacho.biz'], [5725, 'Martyn Caldera', 'martyn.caldera@hotmail.co.uk'], [4037, 'Abigail Foster', 'abigail.foster@taylor.net'], [2109, 'Anthony Hauffer', 'anthony.hauffer@tustek.org'], [3283, 'Roger Gerlach', 'roger.gerlach@may-mitchell.co.uk'], [6181, 'Christopher Poirier', 'christopher.poirier@hotmail.com'], [3251, 'Branka Traore', 'branka.traore@jacobi.com'], [3443, 'Briana Etienne', 'briana.etienne@bishop-coates.com'], [2988, 'Brandon Olson', 'brandon.olson@prince-moreno.net'], [1673, 'Celia Ungaretti', 'celia.ungaretti@fox.com'], [2085, 'Barbara Guibert', 'barbara.guibert@casares-sanches.com'], [1672, 'Daniela Bell', 'daniela.bell@vollbrecht.org'], [4426, 'Phyllis Gurule', 'phyllis.gurule@rodriguez-pham.com'], [3211, 'Alejandra Joseph', 'alejandra.joseph@escalante-abrego.biz'], [3190, 'Giulio Bohlander', 'giulio.bohlander@dbmail.com'], [2081, 'Amber Weller', 'amber.weller@tosu.de'], [3185, 'Mary Etzold', 'mary.etzold@conrad-harrison.com'], [5044, 'Amy Georges', 'amy.georges@aedel.de'], [1883, 'Jennifer Giraud', 'jennifer.giraud@yahoo.com'], [2067, 'Rosie Peltier', 'rosie.peltier@voila.fr'], [4604, 'Oreste Ward', 'oreste.ward@proyectos.com'], [3203, 'Luis Edwards', 'luis.edwards@live.com'], [5758, 'Fiona Griffiths', 'fiona.griffiths@curatoli-verdone.it'], [5755, 'Sally Raya', 'sally.raya@tarchetti.it'], [5150, 'Carolyn Charpentier', 'carolyn.charpentier@hotmail.de'], [4181, 'Nathalie Alvarez', 'nathalie.alvarez@live.com'], [3006, 'Abril Lowery', 'abril.lowery@novak.net'], [4865, 'Ottone Sullivan', 'ottone.sullivan@hussain-kaur.com'], [2862, 'Esmeralda Sollima', 'esmeralda.sollima@visintini.it'], [6070, 'Silvia Gierschner', 'silvia.gierschner@tele2.it'], [5300, 'Kyle Cunningham', 'kyle.cunningham@voila.fr'], [3486, 'Anel Carpentier', 'anel.carpentier@klein-joseph.org'], [5989, 'Sarah Davis', 'sarah.davis@david.com'], [2849, 'Luke Klein', 'luke.klein@hotmail.com'], [1612, 'Gudrun Mueller', 'gudrun.mueller@dillon-fulton-fran.info'], [3307, 'Melissa Canali', 'melissa.canali@libero.it'], [5288, 'Gabrielle Ayala', 'gabrielle.ayala@hotmail.com'], [6026, 'Laura Buckley', 'laura.buckley@ryan.org'], [2212, 'Wilfrido Lorch', 'wilfrido.lorch@leclerc.fr'], [4591, 'Robert Bruder', 'robert.bruder@manzoni.com'], [2771, 'Phillip Soliz', 'phillip.soliz@outlook.com'], [5682, 'Alex Smith', 'alex.smith@fournier.fr'], [5368, 'Jessica Martinez', 'jessica.martinez@salieri.com'], [3706, 'Kelly Carter', 'kelly.carter@hotmail.com.au'], [4034, 'Ahmad Sosa', 'ahmad.sosa@gmail.com'], [3209, 'Toralf Iglesias', 'toralf.iglesias@borrego.com'], [2384, 'Irene Anichini', 'irene.anichini@yahoo.com'], [3074, 'Alicia Cainero', 'alicia.cainero@yahoo.com'], [2031, 'Tracey Laboy', 'tracey.laboy@bell.biz'], [5873, 'Donald Reynaud', 'donald.reynaud@green.edu.au'], [5501, 'Venancio Stadelmann', 'venancio.stadelmann@gmx.de'], [3489, 'Seth Olivier', 'seth.olivier@yahoo.com'], [4210, 'Dennis Boaga', 'dennis.boaga@gmx.de'], [6151, 'Pedro Wilson', 'pedro.wilson@outlook.com'], [6047, 'Roland Wehmann', 'roland.wehmann@gmail.com'], [5445, 'Hector Morrison', 'hector.morrison@web.de'], [5493, 'Evangalos Peters', 'evangelos.peters@sanchez.net'], [6036, 'Gerardo Hamilton', 'gerardo.hamilton@libero.it'], [2368, 'Meta Cuzzocrea', 'meta.cuzzocrea@howard-jensen.org'], [1501, 'Alma Raymond', 'alma.raymond@tin.it'], [4351, 'Kathleen Solimena', 'kathleen.solimena@wallace.biz'], [3096, 'Zacharie Cordier', 'zacharie.cordier@gmail.com'], [6162, 'Riza Techer', 'riza.techer@yahoo.com'], [1433, 'Enrique Guillen', 'enrique.guillen@hotmail.co.uk'], [2720, 'Tristan Weeks', 'tristan.weeks@opena.com'], [5251, 'Natalie Lerma', 'natalie.lerma@boucher.com'], [1797, 'Rodney Hamon', 'rodney.hamon@hotmail.com'], [1656, 'Adalberto Lombardo', 'adalberto.lombardo@sfr.fr'], [1346, 'Flores Quirini', 'flores.quirini@gmail.com'], [2989, 'Liam Rolland', 'liam.rolland@yahoo.fr'], [5629, 'Ian Kelly', 'ian.kelly@reeves.com.au'], [3456, 'Mehdi Todd', 'mehdi.todd@web.de'], [3229, 'Rembrandt Liguori', 'rembrandt.liguori@davis.org'], [2277, 'Irmi Schweitzer', 'irmi.schweitzer@ozuna.com'], [1276, 'Giovanna Stein', 'giovanna.stein@hotmail.com.au'], [3694, 'Frida Villarreal', 'frida.villarreal@hotmail.de'], [2260, 'Steffen Mangold', 'steffen.mangold@paul.net'], [5374, 'Elias Marin', 'elias.marin@herve.net'], [4420, 'Roman Gonzalez', 'roman.gonzalez@anderson.net'], [3849, 'Reina Middleton', 'reina.middleton@hotmail.com'], [1638, 'Sandro Moran', 'sandro.moran@hotmail.de'], [5230, 'Gebhard Thanel', 'gebhard.thanel@gmail.com'], [1763, 'Antonio Bolander', 'antonio.bolander@gmail.com'], [4323, 'Karl Trevisani', 'karl.trevisani@anguissola.it'], [5256, 'Silvester Brooke', 'silvester.brooke@franco.org'], [4836, 'Octavio Pratt', 'octavio.pratt@web.de'], [5981, 'Rebecca Green', 'rebecca.green@hotmail.com'], [1463, 'Pierina Gaggini', 'pierina.gaggini@free.fr'], [4577, 'Robin Schleich', 'robin.schleich@rogers-barrera.biz'], [4951, 'Sergio Abbott', 'sergio.abbott@moore.net.au'], [3567, 'Gilberto Evans', 'gilberto.evans@salcedo-archuleta.net'], [5475, 'James Elliott', 'james.elliott@gmail.com'], [2114, 'Timoteo Bolnbach', 'timoteo.bolnbach@laboratorios.com'], [5216, 'Tonino Stanley', 'tonino.stanley@hotmail.com'], [1665, 'Margaretha Murialdo', 'margaretha.murialdo@meunier.com'], [6051, 'Bianca Drubin', 'bianca.drubin@libero.it'], [3983, 'Kaylee Hoffmann', 'kaylee.hoffmann@poulain.com'], [2966, 'Teresa Vecellio', 'teresa.vecellio@yahoo.com'], [4478, 'Emine Tacchini', 'emine.tacchini@dbmail.com'], [4238, 'Dawn Cortez', 'dawn.cortez@gmail.com'], [4404, 'Emily Cundari', 'emily.cundari@clark.com.au'], [5666, 'Eleanor Hall', 'eleanor.hall@hotmail.com'], [3691, 'Monique Delgadillo', 'monique.delgadillo@hotmail.co.uk'], [2703, 'Arnaude Warner', 'arnaude.warner@gmail.com'], [4253, 'Jason Howell', 'jason.howell@gmx.de'], [5652, 'Nicole McCullough', 'nicole.mccullough@hall.com'], [3279, 'Rosmarie Esquibel', 'rosmarie.esquibel@bouygtel.fr'], [3233, 'Leonor Ferreira', 'leonor.ferreira@gmail.com'], [3017, 'Brett Jones', 'brett.jones@alice.it'], [3482, 'Nino Bien', 'nino.bien@gmail.com'], [2166, 'Jemma Tate', 'jemma.tate@hotmail.de'], [4146, 'Denis Rogers', 'denis.rogers@poste.it'], [2651, 'Vanessa Khan', 'vanessa.khan@voila.fr'], [1409, 'Victoria Ojeda', 'victoria.ojeda@doehn.com'], [3717, 'Eugenio Martin', 'eugenio.martin@yahoo.de'], [3963, 'Nancy Hayes', 'nancy.hayes@hotmail.com'], [2152, 'Lara Jacob', 'lara.jacob@buckley.org'], [2974, 'Gino Hernandez', 'gino.hernandez@covarrubias.com'], [1268, 'Vicente Tijerina', 'vicente.tijerina@montanariello.com'], [1822, 'Michele Brambilla', 'michele.brambilla@yahoo.com'], [5066, 'Dana Ritacca', 'dana.ritacca@gmail.com'], [5075, 'Mitchell Bachmann', 'mitchell.bachmann@suarez-cruz.com'], [3855, 'Vincenza Key', 'vincenza.key@adams.com'], [5923, 'Salvador Gaillard', 'salvador.gaillard@yahoo.de'], [1867, 'Diane Rogner', 'diane.rogner@kelly.co.uk'], [4376, 'Josette Laine', 'josette.laine@saracino-parisi.net'], [2785, 'Courtney Fiebig', 'courtney.fiebig@doerschner.com'], [5884, 'Hugues Oliver', 'hugues.oliver@stiffel.com'], [2441, 'Espartaco Willis', 'espartaco.willis@hotmail.com'], [1243, 'Flavia Yoder', 'flavia.yoder@hotmail.de'], [3411, 'Casey Flores', 'casey.flores@baggio.org'], [2467, 'Calogero Cross', 'calogero.cross@mercantini.it'], [1850, 'Charles Begum', 'charles.begum@laboratorio-s.info'], [2604, 'Adina Pollard', 'adina.pollard@outlook.com'], [1693, 'Natasha Lara', 'natasha.lara@outlook.com'], [2969, 'Benita Pottier', 'benita.pottier@sfr.fr'], [1544, 'Elizabeth Valdivia', 'elizabeth.valdivia@gmail.com'], [4149, 'Mattia Huet', 'mattia.huet@aol.de'], [3463, 'Geza Howard', 'geza.howard@voila.fr'], [2939, 'Lauren Guillaume', 'lauren.guillaume@sorgatz.org'], [2202, 'Joanna Baxter', 'joanna.baxter@bruce-wright.com'], [1728, 'Tracy Metz', 'tracy.metz@bryant.info'], [5362, 'Birgit Pena', 'birgit.pena@armellini.it'], [4840, 'Elvira Papafava', 'elvira.papafava@france.com'], [2955, 'Ernest Delorme', 'ernest.delorme@dowerg.de'], [5354, 'Christy Grossi', 'christy.grossi@alarcon-tafoya.com'], [3847, 'Hermelinda Olmos', 'hermelinda.olmos@portillo.com'], [3025, 'Herbert Fraser', 'herbert.fraser@holloway.org.au'], [1615, 'Giancarlo Heydrich', 'giancarlo.heydrich@hotmail.com'], [5154, 'Constance Maldonado', 'constance.maldonado@spies.net'], [5005, 'Giampiero Carlier', 'giampiero.carlier@hotmail.com'], [1349, 'Carlo Jacquot', 'carlo.jacquot@hotmail.com'], [3670, 'Gabriel Robles', 'gabriel.robles@yahoo.com'], [2912, 'Gerd Gunpf', 'gerd.gunpf@gmail.com'], [4336, 'Ashley Hellwig', 'ashley.hellwig@libero.it'], [5576, 'Patrick Rosas', 'patrick.rosas@yahoo.com'], [2736, 'Lilla Leonard', 'lilla.leonard@jungfer.com'], [3321, 'Tony Nicolas', 'tony.nicolas@morocco-seddine.net'], [3353, 'Allegra Benedetti', 'allegra.benedetti@milner.com'], [2336, 'Camille Tafuri', 'camille.tafuri@aol.de'], [4369, 'Juan Wheeler', 'juan.wheeler@gilmore.net'], [4464, 'Leopoldo Johnson', 'leopoldo.johnson@hotmail.com.au'], [5626, 'Dietmar Grenier', 'dietmar.grenier@tele2.it'], [1669, 'Leonid Scholl', 'leonid.scholl@collin.com'], [5544, 'Olivie Contarini', 'olivie.contarini@marshall-wright.biz'], [6078, 'Jeffery Lacroix', 'jeffery.lacroix@medina.org'], [2580, 'Edeltraud Chavez', 'edeltraud.chavez@hotmail.com.au'], [3592, 'Mesut Morvan', 'mesut.morvan@gmail.com'], [4914, 'Gregory Jackson', 'gregory.jackson@ortiz.com'], [3848, 'Nicola Rossi', 'nicola.rossi@berry.org'], [2619, 'Eddie Collazo', 'eddie.collazo@gmail.com'], [4605, 'Philippe Gardner', 'philippe.gardner@howells-jones.net'], [6147, 'Roy Fox', 'roy.fox@tejeda.com'], [1383, 'Heiderose Garcia', 'heiderose.garcia@noos.fr'], [1757, 'Claudine Rowley', 'claudine.rowley@industrias.com'], [4390, 'Astrid Roht', 'astrid.roht@fastwebnet.it'], [4165, 'Auguste Burnett', 'auguste.burnett@yahoo.fr'], [4529, 'Pellegrino Allen', 'pellegrino.allen@gmail.com'], [6084, 'Angela Bowers', 'angela.bowers@laboratorios.com'], [2916, 'Michael Descamps', 'michael.descamps@gmail.com'], [2005, 'Heidelore Kennedy', 'heidelore.kennedy@guinizzelli.org'], [2261, 'Lucrezia Koch', 'lucrezia.koch@petit.com'], [5342, 'Ivan Accardo', 'ivan.accardo@web.de'], [4087, 'Karen Avogadro', 'karen.avogadro@yahoo.com'], [2761, 'Laetitia Gallet', 'laetitia.gallet

@aubry.org'], [2908, 'Amico Gosselin', 'amico.gosselin@shaw.info'], [2155, 'Ilaria Bowen', 'ilaria.bowen@hotmail.it'], [5642, 'Marlen Coardi', 'marlen.coardi@colletti.eu'], [6217, 'Rafael Alexander', 'rafael.alexander@jenkins.com'], [4222, 'Herma Gabbana', 'herma.gabbana@hotmail.com'], [4798, 'Zachary Vespucci', 'zachary.vespucci@industrias.com'], [5660, 'Marisol Richard', 'marisol.richard@gmail.com'], [5000, 'Joseph Glover', 'joseph.glover@laposte.net'], [2809, 'Lauretta Neal', 'lauretta.neal@hotmail.de'], [3349, 'Simone Sauvage', 'simone.sauvage@yahoo.com'], [3647, 'Ramona Brooks', 'ramona.brooks@kensy.de'], [5745, 'Gioele Schwital', 'gioele.schwital@moody.com'], [2583, 'Elena Jimenez', 'elena.jimenez@yahoo.com.au'], [5441, 'Pam Vargas', 'pam.vargas@guillot.net'], [2090, 'Alison Morales', 'alison.morales@lewis.com'], [3386, 'Marc Aumann', 'marc.aumann@holland.com'], [1816, 'Carla Butler', 'carla.butler@yahoo.com'], [4718, 'Dionigi Bruce', 'dionigi.bruce@industrias.com'], [1677, 'Frank Henry', 'frank.henry@yahoo.com'], [4218, 'Brent Pons', 'brent.pons@proyectos.net'], [4548, 'Patricia Liebelt', 'patricia.liebelt@baca.org'], [1431, 'Micheletto Sykes', 'micheletto.sykes@thompson-thompson.com'], [2830, 'Rose Blanc', 'ros.e.blanc@grupo.com'], [4400, 'Vincent Singh', 'vincent.singh@gmail.com'], [5773, 'Jamie Peacock', 'jamie.peacock@interiano-nordio.com'], [4161, 'Giorgia Dijoux', 'giorgia.dijoux@paul.com'], [2062, 'Serena Pia centini', 'serena.piacentini@yahoo.de'], [3458, 'Dimitri Posada', 'dimitri.posada@gmail.com'], [3174, 'Alexa Barillaro', 'alexa.barillaro@ramos.net'], [5565, 'Megan Kuhl', 'megan.kuhl@despacho.com'], [1906, 'Kaitlyn Farinelli', 'kaitlyn.farinelli@outlook.com'], [5479, 'Jonathan Castellitto', 'jonathan.castellitto@libero.it'], [4320, 'Lisa Terragni', 'lisa.terragni@proyectos.com'], [4213, 'Olivier Petrucci', 'olivier.petrucci@letta-naurica.eu'], [3373, 'Jos Trobbiani', 'jos.trobbiani@samson.com'], [4184, 'Mirjam Dawson', 'mirjam.dawson@yahoo.com'], [5472, 'Javier Tessier', 'javier.tessier@mclaughlin.biz'], [5559, 'Mirjana Collins', 'mirjana.collins@hotmail.com'], [5419, 'Vanessa Tschentscher', 'vanessa.tschentscher@mtichell.com'], [4171, 'Antonina Cobb', 'antonina.cobb@gibson.org'], [1302, 'Bernard Durand', 'bernard.durand@royer.com'], [3520, 'Corey Greggio', 'corey.greggio@johnson.org.au'], [6010, 'Rodolfo Couturier', 'rodolfo.couturier@tim.it'], [4617, 'Pierre Cardenas', 'pierre.cardenas@luzi-bonomo.eu'], [1417, 'Romeo Surian', 'romeo.surian@batista.com'], [4770, 'Ivo Moreno', 'ivo.moreno@hotmail.com'], [4328, 'Federico Marie', 'federico.marie@journan.fr'], [5171, 'Maggie Boito', 'maggie.boito@yahoo.com'], [4394, 'Brianna Hanson', 'brianna.hanson@stanley.com'], [1794, 'Heiko Bourgeois', 'heiko.bourgeois@togliatti.com'], [3510, 'Benito Gritti', 'benito.gritti@gmail.com'], [2487, 'Mitzy Junk', 'mitzy.junk@zaragoza.com'], [2215, 'Samantha Lowe', 'samantha.lowe@praga.it'], [4136, 'Ernesto Nergen', 'ernesto.nergen@middleton.co.uk'], [3723, 'Melinda Canova', 'melinda.canova@orange.fr'], [5842, 'Mohammad Hartung', 'mohammad.hartung@hotmail.com'], [3289, 'Ben Maillot', 'ben.maillot@ramos.net'], [1249, 'Isabel Pruschke', 'isabel.pruschke@gmail.com'], [5409, 'Linda Franceschi', 'linda.franceschi@richards.net'], [2951, 'Olivia Striebitz', 'olivia.striebitz@gmail.com'], [3432, 'Penny Watson', 'penny.watson@matthews.net'], [2297, 'Manfred James', 'manfred.james@davies.org'], [6161, 'Hazel Pierce', 'hazel.pierce@outlook.com'], [4603, 'Alf Bates', 'alf.bates@duunn.com'], [5818, 'Scott Gough', 'scott.gough@industrias.com'], [4270, 'Konrad Briones', 'konrad.briones@fierno.com'], [6007, 'Mathilde Kruschwitz', 'mathilde.kruschwitz@yahoo.com'], [4084, 'Irma Gemito', 'irma.gemito@googlemail.com'], [1340, 'Antoine Guyon', 'antoine.guyon@yahoo.com'], [5675, 'William Gibilisco', 'william.gibilisco@gaillard.com'], [2010, 'Judith Butte', 'judith.butte@fischer-aughn.info'], [6201, 'Joanne Henderson', 'joanne.henderson@gmail.com'], [1791, 'Ludovica Arellano', 'ludovica.arellano@morandi-argento.com'], [3314, 'Catherine Gray', 'catherine.gray@hotmail.com'], [5852, 'Franco Fioravanti', 'franco.fioravanti@barry-gill.com.au'], [4618, 'Heather Zimmer', 'heather.zimmer@bluemel.com'], [2618, 'Hortense Taylor', 'hortense.taylor@chambers.biz'], [4967, 'Livia Ernst', 'livia.ernst@colas.fr'], [4594, 'Toby Etzler', 'toby.etzler@industrias.com'], [2080, 'Ennio Leoncavallo', 'ennio.leoncavallo@sanchez.org'], [2793, 'Philippine Vaca', 'philippine.vaca@hotmail.com'], [4890, 'Rhonda Phillips', 'rhonda.phillips@yahoo.com'], [3528, 'Alphonse Nitto', 'alphonse.nitto@voila.fr'], [5810, 'Karina Arreola', 'karina.arreola@yahoo.com.au'], [5720, 'Denise Wood', 'denise.wood@antegna.com'], [4593, 'Burghard Rosenow', 'burghard.rosenow@hotmail.com'], [3186, 'Friedl Powell', 'friedl.powell@laposte.net'], [5136, 'Benjamin Schultz', 'benjamin.schultz@rivero-davila.com'], [2646, 'Rachel Ulibarri', 'rachel.ulibarri@schueler.com'], [3460, 'Silvano Blanchet', 'silvano.blanchet@yahoo.com'], [4345, 'Kurt Sauer', 'kurt.sauer@almanza.biz'], [2690, 'Nayeli Gehringer', 'nayeli.gehringer@hotmail.com'], [4025, 'Imelda Guerin', 'imelda.guerin@stumpf.net'], [4524, 'Zbigniew Holmes', 'zbigniew.holmes@stiffel.net'], [3978, 'Adela Alfonsi', 'adela.alfonsi@despacho.com'], [3910, 'Hilary Clark', 'hilary.clark@libero.it'], [5825, 'Kornelius Molina', 'kornelius.molina@heuser.com'], [3564, 'Christian Anderson', 'christian.anderson@tim.it'], [2396, 'Francisco Dorsey', 'francisco.dorsey@tiscali.fr'], [2140, 'Walli Albert', 'walli.albert@dussen.com'], [3079, 'Siegrun Mota', 'siegrun.mota@matthews-montoya.org.au'], [2017, 'Alberico Herve', 'alberico.herve@beer.net'], [5444, 'Serafina Bonnin', 'serafina.bonnin@mendes.com'], [2216, 'Marcelle Bendor', 'marcelle.bendor@edwards-robinson.info'], [1241, 'Hartmut Robertson', 'hartmut.robertson@thomas.edu.au'], [1484, 'Herlinde George', 'herlinde.george@free.fr'], [3475, 'Pamela Payne', 'pamela.payne@yahoo.com'], [4333, 'Gastone Beyer', 'gastone.beyer@stewart-sanchez.edu'], [5657, 'Lydia Berengario', 'lydia.berengario@aol.de'], [2527, 'Lucy Davidson', 'lucy.davidson@tele2.it'], [5967, 'Frank Wade', 'frank.wade@gute.org'], [5698, 'Heidi Barker', 'heidi.barker@gmail.com'], [3331, 'Arnaldo Ocasio', 'arnaldo.ocasio@hotmail.com'], [3841, 'Francis Bernier', 'francis.bernier@rodriguez.com'], [6105, 'Johanne Strong', 'johanne.strong@yahoo.de'], [3895, 'Debra Langlois', 'debra.langlois@schmiedecke.com'], [1891, 'Reingard Kams', 'reingard.kams@sagese.eu'], [1281, 'Mauro Ortiz', 'mauro.ortiz@oquendo.net'], [4186, 'Griseld Carranza', 'griselda.carranza@krause.de'], [2511, 'Janet Madrigal', 'janet.madrigal@gmail.com'], [4272, 'Jenna Day', 'jenna.day@reed.com'], [3533, 'Modesto Wright', 'modesto.wright@pareto.com'], [3618, 'Alice Wall', 'alice.wall@zaccardo.it'], [2177, 'Brenda Sontag', 'brenda.sontag@solari.com'], [4102, 'Sergius Mace', 'sergius.mace@palazzo.com'], [6089, 'Antonia Kline', 'antonia.kline@gmail.com'], [4701, 'Maurice Gumprich', 'maurice.gumprich@griffin.net.au'], [5645, 'Daniel Anders', 'daniel.anders@yahoo.com'], [4499, 'Zaira Cattaneo', 'zaira.cattaneo@yahoo.com'], [3365, 'Alessio Bruneau', 'alessio.bruneau@hotmail.co.uk'], [2290, 'Gerdi Guerrero', 'gerdi.guerrero@tele2.fr'], [1890, 'Brigitte Stiebitz', 'brigitte.stiebitz@gmail.com'], [1304, 'June Sinisi', 'june.sinisi@manzoni-giannini.it'], [3799, 'Carl Macias', 'carl.macias@yahoo.com'], [2749, 'Veronica Trapp', 'veronica.trapp@tiscali.it'], [2028, 'Frances Garnier', 'frances.garnier@aporti-guidotti.org'], [2383, 'Terri Iannelli', 'terri.iannelli@hotmail.com'], [5721, 'Odette Moore', 'odette.moore@hall.edu'], [1394, 'Mercedes Trujillo', 'mercedes.trujillo@aol.de'], [3451, 'Babette Dumas', 'babette.dumas@brady.net'], [5512, 'Jerry Santiago', 'jerry.santiago@soprano-ferragn-i.it'], [2453, 'Ronald Simpson', 'ronald.simpson@ovadia.net'], [6055, 'Emmanuel Dominguez', 'emmanuel.dominguez@gmail.com'], [5704, 'Christiane Ricciardi', 'christiane.ricciardi@zavala.com'], [3707, 'Ingolf Zamora', 'ingolf.zamora@yahoo.com.au'], [2205, 'Rita Sharp', 'rita.sharp@beard-scott.edu.au'], [5976, 'Geraldine Cabrera', 'geraldine.cabrera@green-smith.info'], [3290, 'Chantal Armas', 'chantal.armas@jenkin-s.net'], [4257, 'Louis Gilbert', 'louis.gilbert@gmail.com'], [4588, 'Michelle Collier', 'michelle.collier@peruzzi.it'], [4785, 'Guillermina Nicholson', 'guillermina.nicholson@grondin.fr'], [3033, 'Bertrand Hopkins', 'bertrand.hopkins@gmail.com'], [2779, 'Alonso Palomo', 'alonso.palomo@gehringer.de'], [2131, 'Tina Rodriguez', 'tina.rodriguez@web.de'], [6229, 'Eva Francis', 'eva.francis@parker.org'], [3429, 'Otta viuo Rust', 'ottavio.rust@respi.com'], [1613, 'Amelia Olvera', 'amelia.olvera@hotmail.com'], [4734, 'Paul Palladio', 'paul.palladio@mclaughlin.net.au'], [4439, 'Bernardo Wieloch', 'bernardo.wieloch@budig.com'], [1441, 'Umberto Hicks', 'umberto.hicks@hill.com'], [5799, 'Janko Bishop', 'janko.bishop@tim.it'], [1419, 'Jessika Meunier', 'jessika.meunier@gmx.de'], [3059, 'Isaac Benard', 'isaac.benard@googlemail.com'], [3224, 'Alexandrie Lambert', 'alexandrie.lambert@yahoo.com'], [3181, 'Roberto Guyot', 'roberto.guyot@bennett.com'], [6061, 'Giulietta Lucas', 'giulietta.lucas@club.com'], [4466, 'Damaris Pininfarina', 'damaris.pininfarina@gmail.com'], [6081, 'Fabrizia Renard', 'fabrizia.renard@higgins-marks.com'], [2159, 'Camillo Dehmel', 'camillo.dehmel@brown-campbell.com'], [3324, 'Ruggero Parpinel', 'ruggero.parpinel@gmail.com'], [3176, 'Stephen Pratesi', 'stephen.pratesi@hotmail.com'], [5874, 'Ludger Naccari', 'ludger.naccari@ingram.biz'], [5110, 'Meryem Tapia', 'meryem.tapia@yahoo.com'], [5490, 'Jasmine Juvara', 'jasmine.e.juvara@yahoo.com.au'], [1922, 'Victoire Travaglia', 'victoire.travaglia@zaccchia.com'], [4398, 'Adrian Noel', 'adrian.noel@perrin.com'], [4754, 'Marvin Howe', 'marvin.howe@gmail.com'], [3309, 'Hanne Kidd', 'hanne.kidd@yahoo.com'], [1244, 'Donatello Millet', 'donatello.millet@fritsch.net'], [2892, 'Eloisa Pascarella', 'eloisa.pascarella@gmail.com'], [2777, 'Erica Zambrano', 'erica.zambrano@gmail.com'], [4952, 'Paula Canales', 'paula.canales@gilmore-guerrero.com'], [5144, 'Claudia Bernard', 'claudia.bernard@mazzanti.it'], [5487, 'Paulina Miller', 'paulina.miller@lewis-barker.com'], [1932, 'Kathrin Godfrey', 'kathrin.godfrey@tele2.fr'], [3364, 'Pina Passalacqua', 'pina.passalacqua@yahoo.co.uk'], [5165, 'Julien Miles', 'julien.miles@stevens-rose.com'], [3858, 'Suzanne Benigni', 'suzanne.benigni@mennea-morlacchi.com'], [4666, 'Annalisa Pugh', 'annalisa.pugh@hotmail.fr'], [2896, 'Mathew Maestas', 'mathew.maestas@casas-garrido.com'], [4520, 'Laszlo Vallee', 'laszlo.vallee@verdiere.com'], [5726, 'Helen Polanco', 'helen.polanco@hotmail.com'], [5661, 'Mateo Sorrentino', 'mateo.sorrentino@calderon.com'], [4765, 'Eric Barnes', 'eric.barnes@barrett-winter.com'], [3710, 'Dragica West', 'dragica.west@yahoo.com.au'], [4925, 'Antje Voisin', 'antje.voisin@gmail.com'], [1533, 'Stanley Mills', 'stanley.mills@parker-lee.biz'], [5264, 'Brandy Meraz', 'brandy.meraz@yahoo.com'], [4606, 'Conor Reinhardt', 'conor.reinhardt@yahoo.com.au'], [2043, 'Gabi Blot', 'gabi.blot@lewis-russell.org'], [3871, 'Alvaro Lemus', 'alvaro.lemus@yahoo.com'], [6205, 'Patrizia Wilms', 'patrizia.wilms@dbmail.com'], [3193, 'Metin Coulon', 'metin.coulon@ashley.com'], [1810, 'Reece Valencia', 'reece.valencia@laboy-palomo.com'], [4797, 'Troy Schmiedecke', 'troy.schmiedecke@picard.com'], [1487, 'Wendy Bustos', 'wendy.bustos@libero.it'], [3474, 'Guglielmo Caldwell', 'guglielmo.caldwell@loeffler.com'], [5771, 'Travis Cowan', 'travis.cowan@turner.info'], [6074, 'Brandi Abbagnale', 'brandi.abbagnale@tele2.fr'], [1568, 'Hanno Gollum', 'hanno.gollum@smith.com'], [4263, 'Jutta Burke', 'jutta.burke@elliott.com'], [5850, 'Dorothe Paz', 'dorothe.paz@fiebig.de'], [5619, 'Desiree Huerta', 'desiree.huerta@morris.com'], [1904, 'Ilse Bray', 'ilse.bray@venturi.it'], [5388, 'Pierpaolo Scaduto', 'pierpaolo.scaduto@sanders-gibson.com.au'], [2150, 'Amedeo Bradford', 'amedeo.bradford@yahoo.com'], [2653, 'Julia Ali', 'julia.ali@yahoo.com'], [2687, 'Catalina Pizzetti', 'catalina.pizzetti@tim.it'], [3779, 'Dylan Porzio', 'dylan.porzio@yahoo.de'], [1494, 'Iris Trincavelli', 'iris.trincavelli@tiscali.fr'], [4129, 'Johnathan Guidotti', 'johnathan.guidotti@cox-sanchez.net'], [3035, 'Walburga Vollbrecht', 'walburga.vollbrecht@aol.de'], [2069, 'Kendra David', 'kendra.david@allan-morton.com'], [5469, 'Aleksandr Weiss', 'aleksandr.weiss@blanc.org'], [3903, 'Cornelio Guardado', 'cornelio.guardado@gmail.com'], [4083, 'Ferdinand B lackburn', 'ferdinand.blackburn@christensen.net'], [4003, 'Bill Gallagher', 'bill.gallagher@green.com'], [3565, 'Leonard Harvey', 'leonard.harvey@hotmail.com'], [1640, 'Milan Montenegro', 'milan.montenegro@lange-rn.com'], [4769, 'Philip Perez', 'philip.perez@hotmail.co.uk'], [3398, 'Kathryn Hentschel', 'kathryn.hentschel@shaw.com'], [1865, 'Xavier Samson', 'xavier.samson@baggio.it'], [5690, 'Hartwig Roberts', 'hartwig.ig.roberts@saraceno.net'], [3814, 'Kimberly Chan', 'kimberly.chan@gmail.com'], [2942, 'Lorraine Crawford', 'lorraine.crawford@gmail.com'], [5095, 'Maximiliano Knowles', 'maximiliano.knowles@gmail.com'], [2878, 'Gabriella Avila', 'gabriella.avila@gmail.com'], [3697, 'Rico Mortati', 'rico.mortati@faivre.fr'], [4744, 'Gail Petrocelli', 'gail.petrocelli@yahoo.com'], [2104, 'Liberto Niemeier', 'liberto.niemeier@germ-ano-dibiassi.it'], [4883, 'Luciana Petitjean', 'luciana.petitjean@patterson-cole.biz'], [4807, 'Ron Cavazos', 'ron.cavazos@yahoo.com'], [1372, 'Aldo Cibirin', 'aldo.cibirin@gmx.de'], [3666, 'Alexandre Serrano', 'alexandre.serrano@yahoo.co.uk'], [2066, 'Lori Fagotto', 'lori.fagotto@hill-anderson.com'], [1834, 'Perla Figueroa', 'perla.figueroa@live.com'], [5853, 'Reginald Hughes', 'reginald.hughes@gmail.com'], [2219, 'Uriel Holt', 'uriel.holt@gmail.com'], [4148, 'Stefano Lemonnier', 'stefano.lemonnier@binner.com'], [4477, 'Madeleine Cooper', 'madeleine.cooper@poste.it'], [1786, 'Alban Barbier', 'alban.barbier@ortiz.org'], [3455, 'Alfonso Finetti', 'alfonso.finetti@quiroz.com'], [5707, 'Ryan Kim', 'ryan.kim@noos.fr'], [5470, 'Augustin Wohlgenut', 'augustin.wohlgenut@watson.info'], [3274, 'Rolando Bibi', 'rolando.bibi@tin.it'], [4613, 'Berthold Randall', 'berthold.randall@mcdonald.com'], [4465, 'Genaro Briand', 'genaro.briand@sfr.fr'], [1652, 'Rudolf Jenkins', 'rudolf.jenkins@hotmail.com'], [2100, 'Pauline Ackermann', 'paulin

e.ackermann@arredondo-roque.com'], [4321, 'Folker Swanson', 'folker.swanson@rodrigues.fr'], [4168, 'Ashleigh Montez', 'ashleigh.montez@perry-schneider.com'], [3733, 'Liliana Barber', 'liliana.barber@libero.it'], [2594, 'Caridad Carr', 'caridad.carr@huhn.com'], [3909, 'Lolita Borrego', 'lolita.borrego@gmail.com'], [5179, 'Gunar Patrick', 'gunar.patrick@leroy.com'], [2228, 'Leokadia Verdier', 'leokadia.verdier@hotmail.com.au'], [6222, 'Hanna Reid', 'hanna.reid@salas.org'], [2354, 'Edelgard Kreusel', 'edelgard.kreusel@hotmail.com'], [1725, 'Svenja Perrin', 'svenja.perrin@berger.com'], [2960, 'Pierluigi Taccola', 'pie luigi.taccola@gmx.de'], [2077, 'Cilli Watts', 'cilli.watts@yahoo.com'], [5662, 'Emiliano Wells', 'emiliano.wells@gmx.de'], [3481, 'April Davids', 'april.davids@fisher.info'], [2565, 'Dolores Didier', 'dolor es.didier@yahoo.com.au'], [4436, 'Sylvia Bonomo', 'sylvia.bonomo@poulain.com'], [5072, 'Nelly Marsh', 'nelly.marsh@hotmail.it'], [5200, 'Beth Miniati', 'beth.miniati@hudson.net'], [4502, 'Renee Marino', 'ren ee.merino@gmail.com'], [4945, 'Hailey Russell', 'hailey.russell@jourdan.org'], [3012, 'Raphaela Gotthard', 'raphaela.gotthard@outlook.com'], [6094, 'Pomponio Comencini', 'pomponio.comencini@hotmail.co.uk'], [3815, 'Leone Richardson', 'leone.richardson@yahoo.com'], [4838, 'Valentine Acosta', 'valentine.acosta@peters.net'], [3469, 'Ricciotti Laureano', 'ricciotti.laureano@parsons.com'], [4874, 'Jacques Heidrich', 'jacques.heidrich@zamudio-chapa.com'], [1792, 'Anne Snow', 'anne.snow@gmail.com'], [4356, 'Sonia Volta', 'sonia.volta@gmx.de'], [6196, 'Dale Leblanc', 'dale.leblanc@stewart.com'], [3578, 'Alejandro Cruz', 'a lejandro.cruz@hotmail.com'], [4491, 'Elsa Cugia', 'elsa.cugia@yahoo.com'], [3054, 'Geronimo Pechel', 'geronimo.pechel@hotmail.com'], [1968, 'Melania Klemm', 'melania.klemm@seidel.com'], [5869, 'Katherine Cooley', 'katherine.cooley@yahoo.com'], [5858, 'Guillermo Larsen', 'guillermo.larsen@yahoo.com'], [5203, 'Jill Barrios', 'jill.barrios@gmail.com'], [2577, 'Austin Musatti', 'austin.musatti@tin.it'], [3216, 'Jac into Rose', 'jacinto.rose@yahoo.com'], [2037, 'Lucie Redi', 'lucie.reidi@googlemail.com'], [3867, 'Tania Chapman', 'tania.chapman@remy.com'], [5109, 'Alexandria Best', 'alexandria.best@yahoo.co.uk'], [1376, 'Edward Scott', 'edward.scott@yahoo.com'], [5260, 'Jake Wiley', 'jake.wiley@outlook.com'], [2350, 'Claudio Reuter', 'claudio.reuter@morgan-collins.org.au'], [2269, 'Laure Traetta', 'laure.traetta@kay-perkin s.co.uk'], [3650, 'Magarete Valentin', 'magarete.valentin@yahoo.com'], [3934, 'Paige Hill', 'paige.hill@jones.org.au'], [4048, 'Fulvio Curiel', 'fulvio.curiel@reynolds.com'], [4266, 'Thibaut Camarillo', 'thi baut.camarillo@hotmail.com'], [3863, 'Romana Blanchard', 'romana.blanchard@googlemail.com'], [4922, 'Thibault Savorgnan', 'thibault.savorgnan@butler.biz'], [4517, 'Alexander Pausini', 'alexander.pausini@gonz alez.info'], [3259, 'Pierangelo Scholtz', 'pierangelo.scholtz@lefevre.com'], [3123, 'Emmanuelle Keller', 'emmanuelle.keller@morton-clarke.com'], [3870, 'Isabella Deleon', 'isabella.deleon@web.de'], [2996, 'T imothy Weitzel', 'timothy.weitzel@gmail.com'], [2993, 'Alfredo Ingram', 'alfredo.ingram@hotmail.com'], [4556, 'Francesca Kade', 'francesca.kade@luna-rael.org'], [2326, 'Tonia Coleman', 'tonia.coleman@hotmail .com'], [4109, 'Gottfried Barbe', 'gottfried.barbe@hotmail.com'], [5739, 'Jeffrey Trommler', 'jeffrey.trommler@yahoo.com'], [5887, 'Giacobbe Owens', 'giacobbe.owens@huet.fr'], [1651, 'Conchita Pagliaro', 'c onchita.pagliaro@googlemail.com'], [1628, 'Kevin Delahaye', 'kevin.delahaye@ellis-wilson.com'], [2642, 'Rouven Ovadia', 'rouven.ovadia@gmail.com'], [6020, 'Piermaria Laporte', 'piermaria.laporte@wesack.com'], [4747, 'Karolin Johann', 'karolin.johann@hotmail.com'], [5534, 'Steven Folliero', 'steven.folliero@noos.fr'], [4790, 'Livio Rosselli', 'livio.rosselli@green.co.uk'], [4632, 'Gaetano Leonardi', 'gaetano. leonardi@despacho.com'], [1952, 'Cynthia Robinson', 'cynthia.robinson@gilles.net'], [4388, 'Eleni Aponte', 'eleni.aponte@club-internet.fr'], [5529, 'Noelia Romo', 'noelia.romo@yahoo.com'], [1316, 'Evelyn Hay nes', 'evelyn.haynes@clark.info'], [1670, 'Katharine Zaccagnini', 'katharine.zaccagnini@hotmail.com'], [4909, 'Lea Merle', 'lea.merle@stey.de'], [3401, 'Marta Huynh', 'marta.huynh@hernandez.info'], [4619, 'A mador Riley', 'amador.riley@hotmail.com'], [4722, 'Julie Coles', 'julie.coles@hotmail.fr'], [2188, 'Beverley Telesio', 'beverley.telesio@regnier.fr'], [5497, 'Corinne Mitschke', 'corinne.mitschke@hauffer.ne t'], [5711, 'Valerie Ortega', 'valerie.ortega@tele2.it'], [4076, 'Carlos Delle', 'carlos.delle@hicks.com'], [1847, 'Christina Besnard', 'christina.besnard@delorme.com'], [2816, 'Rosario Jacques', 'rosario.ja cques@yahoo.com'], [5019, 'Celal Fischer', 'celal.fischer@aol.de'], [4869, 'Leonardo Guzman', 'leonardo.guzman@andreoizzi-abba.com'], [3613, 'Clinton Mason', 'clinton.mason@yahoo.com'], [3069, 'Loretta Malle t', 'loretta.mallet@sansoni-toscani.it'], [2601, 'Lorenzo Roero', 'lorenzo.roero@hotmail.co.uk'], [2133, 'Lando Gutierrez', 'lando.gutierrez@heser.com'], [3164, 'Elaine Patterson', 'elaine.patterson@rose-mor ris.org'], [3897, 'Marcella Govoni', 'marcella.govoni@barkholz.net'], [5663, 'Ettore Amaldi', 'ettore.amaldi@germain.com'], [3879, 'Rufino Gallegos', 'rufino.gallegos@webster-newton.co.uk'], [5900, 'Georges Richards', 'georges.richards@macdonald.com'], [2272, 'Janice Infantino', 'janice.infantino@hotmail.com'], [5436, 'Svetlana Montanariello', 'svetlana.montanariello@putz.de'], [3676, 'Agnolo Cedillo', 'agnolo. cedillo@laboratorios.biz'], [4676, 'Hannah Jockel', 'hannah.jockel@ceravolo-tonisto.eu'], [1563, 'Yvonne Stout', 'yvonne.stout@sims.edu'], [4349, 'Anouk Pace', 'anouk.pace@gmail.com'], [1658, 'Esteban Valde z', 'esteban.valdez@hotmail.com'], [2552, 'Angelina Davies', 'angelina.davies@hotmail.com'], [4847, 'Anastasia Cadena', 'anastasia.cadena@live.com'], [1553, 'Hansjoachim Lettiere', 'hansjoachim.lettierre@hall -mcdaniel.net'], [2632, 'Yeni Tejada', 'yeni.tejada@holson.edu.au'], [1908, 'Dulce Adinolfi', 'dulce.adinolfi@hotmail.com'], [5801, 'Sean Mielcarek', 'sean.mielcarek@gmail.com'], [1935, 'Omar Fonseca', 'omar. fonseca@grupo.org'], [2609, 'Henry Gucci', 'henry.gucci@leleu.fr'], [2740, 'Samira Bottaro', 'samira.bottaro@vasseur.fr'], [1907, 'Korinna Arnold', 'korinna.arnold@zapata-saenz.biz'], [1831, 'Marcela Ferrant e', 'marcela.ferrante@gmx.de'], [5177, 'Douglas Amaya', 'douglas.amaya@favata-brenna.net'], [6179, 'Katherina Barrientos', 'katherina.barrientos@trobbiani.eu'], [2917, 'Brittany Tran', 'brittany.tran@tele2.i t'], [1486, 'Caroline Reeves', 'caroline.reeves@yahoo.com'], [3735, 'Joe Michaud', 'joe.michaud@yahoo.fr'], [3838, 'Marco Vittadello', 'marco.vittadello@gmail.com'], [2430, 'Kim Webb', 'kim.webb@yahoo.com.a u'], [2480, 'Eugenia Zichichi', 'eugenia.zichichi@yahoo.com'], [3431, 'Aurore Rosemann', 'aurore.rosemann@leoncavallo-rusticucci.com'], [1742, 'Giustino Luzi', 'giustino.luzi@rolland.fr'], [4628, 'Anita Gali ndo', 'anita.galindo@gmail.com'], [2385, 'Gunhild Carrillo', 'gunhild.carrillo@elliott-gilbert.org'], [2213, 'Rosa Paul', 'rosa.paul@bien.com'], [5188, 'Marliese Oneal', 'marliese.oneal@gmail.com'], [3418, 'Bethany Hartmann', 'bethany.hartmann@yahoo.com'], [5637, 'Irina Martineau', 'irina.martineau@gmail.com'], [3530, 'Beatriz Moulin', 'beatriz.moulin@pinto.com'], [3731, 'Lucas Austin', 'lucas.austin@gmail.com m'], [3219, 'Carole Nohlmans', 'carole.nohlmans@yahoo.com'], [2392, 'Bozena Schmitt', 'bozena.schmitt@vallet.com'], [5103, 'Aimee Foconi', 'aimee.foconi@outlook.com'], [3992, 'Marzena Carrero', 'marzena.carr ero@roskoth.de'], [2442, 'Raisa Poulain', 'raisa.poulain@fastwebnet.it'], [2913, 'Marcantonio Bennett', 'marcantonio.bennett@hopkins.com.au'], [1631, 'Kenneth Hurst', 'kenneth.hurst@meunier.net'], [4073, 'Ma risela Mende', 'marisela.mende@godoy-enriquez.com'], [4568, 'Blanka Villareal', 'blanka.villareal@monduzzi.net'], [1830, 'Tomislav Stoppani', 'tomislav.stoppani@corporacin.org'], [2623, 'Krystal Ammaniti', 'krystal.ammaniti@barbier.fr'], [4327, 'Deanna Aporti', 'deanna.aporti@boughtel.fr'], [3713, 'Donatella Burns', 'donatella.burns@yahoo.de'], [2389, 'Margrit Hooper', 'margrit.hooper@douglas.com'], [5141, 'Ba ccio Marcacci', 'baccio.marcacci@laposte.net'], [3522, 'Alfio Roman', 'alfio.roman@hotmail.co.uk'], [1334, 'Soledad Soto', 'soledad.soto@hotmail.de'], [4721, 'Josh Cimarosa', 'josh.cimarosa@hotmail.com'], [3 768, 'Nath Rico', 'nath.rico@yahoo.com'], [1639, 'Gary Armani', 'gary.armani@web.de'], [5149, 'Miriam Oquendo', 'miriam.oquendo@smith-watson.biz'], [2153, 'Monica Mohaupt', 'monica.mohaupt@live.com'], [2767, 'Alec Ho', 'alec.ho@parra-esparza.com'], [5124, 'Harry Fouquet', 'harry.fouquet@ibarra-cooper.com'], [2097, 'Arcelia Babati', 'arcelia.babati@orange.fr'], [4796, 'Trinidad Vespa', 'trinidad.vespa@gmail.co m'], [5849, 'Colette Ryan', 'colette.ryan@hotmail.com'], [1495, 'Richard Andre', 'richard.andre@porras.org'], [2462, 'Ellinor Leclercq', 'ellinor.leclercq@gmail.com'], [2280, 'Patricio Guillon', 'patricio.gu illon@web.de'], [1877, 'Kirsten Chindamo', 'kirsten.chindamo@carrillo-wall.com'], [6108, 'Leo Petruzzi', 'leo.petruzzi@yahoo.com'], [4427, 'Margot Tomaselli', 'margot.tomaselli@juncn.org'], [2881, 'Ansaldo M arks', 'ansaldo.marks@patel-odonnell.com'], [2339, 'Regina Morin', 'regina.morin@barcaccia.net'], [5156, 'Karla Medici', 'karla.medici@yahoo.com'], [4750, 'Morgan Irizarry', 'morgan.irizarry@williams-harris. biz'], [4852, 'Natividad Wong', 'natividad.wong@hoevel.com'], [5734, 'Anna Freeman', 'anna.freeman@snyder.net'], [3801, 'Emilia Vivaldi', 'emilia.vivaldi@garnier.fr'], [4474, 'Gerda Cook', 'gerda.cook@tin.i t'], [1992, 'Andrey Faure', 'andrey.faure@frederick-mitchell.info'], [5796, 'Charlotte Duke', 'charlotte.duke@ochoa.org'], [1373, 'Cindy Castillo', 'cindy.castillo@virgilio.it'], [1879, 'Hiltrud Thibault', 'hiltrud.thibault@barbe.com'], [3872, 'Kasimir Jaime', 'kasimir.jaime@gmail.com'], [4561, 'Sabine Lachmann', 'sabine.lachmann@phillips.org'], [5201, 'Andrea Arroyo', 'andrea.arroyo@yahoo.com.au'], [5871, 'Ma tthew Velasquez', 'matthew.velasquez@gmail.com'], [3041, 'Severino Linares', 'severino.linares@angeli.com'], [5864, 'Swantje Lomeli', 'swantje.lomeli@gmx.de'], [3999, 'Marcus Espino', 'marcus.espino@comolli. eu'], [4849, 'Allen Gates', 'allen.gates@gmail.com'], [5461, 'Mauricio Saunders', 'mauricio.saunders@hotmail.co.uk'], [5946, 'Gilbert Cornejo', 'gilbert.cornejo@hotmail.com'], [4508, 'Letizia Gulotta', 'leti zia.gulotta@howard.biz'], [5096, 'Etta Gutknecht', 'etta.gutknecht@leconte.net'], [5443, 'Shelley Perry', 'shelley.perry@gierschneider.de'], [2165, 'Geoffrey Williams', 'geoffrey.williams@poste.it'], [2125, 'Co ry Stewart', 'cory.stewart@voila.fr'], [6141, 'Kristy Bartlett', 'kristy.bartlett@outlook.com'], [5245, 'Gianluca Beck', 'gianluca.beck@green.biz'], [2190, 'Guido Krein', 'guido.krein@gmail.com'], [2243, 'Le if Monnier', 'leif.monnier@quenerini.it'], [3336, 'Jonas Badoer', 'jonas.badoer@kennedy.biz'], [3305, 'Devin Ebert', 'devin.ebert@yahoo.fr'], [6191, 'Slobodan Camiscione', 'slobodan.camiscione@hotmail.com'], [1326, 'Volkmar Gauthier', 'volkmar.gauthier@hotmail.com'], [6159, 'Jordan Rowe', 'jordan.rowe@outlook.com'], [2003, 'Manon Daugherty', 'manon.daugherty@segre.com'], [5881, 'Estela Segovia', 'estela.segovia@vodafone.it'], [1918, 'Giada Celentano', 'giada.celentano@evans.net'], [2157, 'Clayton Salcedo', 'clayton.salcedo@hotmail.co.uk'], [3149, 'Raven Pinto', 'raven.pinto@hotmail.com'], [2444, 'Lewis Lewis', 'lew is.lewis@pisano.net'], [6207, 'Rochus Rousseau', 'rochus.rousseau@nguyen.com'], [5580, 'Belinda Gamez', 'belinda.gamez@leleu.com'], [5942, 'Angelo Lozano', 'angelo.lozano@macias.com'], [1592, 'Jorge Monti', 'jorge.monti@clement.fr'], [2344, 'Ricky Harrell', 'ricky.harrell@martin.edu'], [5995, 'Adriano Wulf', 'adriano.wulf@schmidtke.net'], [5022, 'Pompeo Bourdon', 'pompeo.bourdon@gmail.com'], [5991, 'Kristie Gar rett', 'kristie.garrett@wernecke.com'], [4020, 'Lodovico Conley', 'lodovico.conley@yahoo.com'], [6028, 'Jean Berthelot', 'jean.berthelot@oscura.it'], [2790, 'Tadeusz Medina', 'tadeusz.medina@jackson-clark.ne t'], [4033, 'Ruby Carriera', 'ruby.carriera@hotmail.it'], [1261, 'Theres Antonioni', 'theres.antonioni@hotmail.com'], [1979, 'Gloria Morton', 'gloria.morton@outlook.com'], [2122, 'Graziano Jensen', 'grazian o.jensen@gmail.com'], [4506, 'Gianpaolo Massey', 'gianpaolo.massey@bouvier.com'], [3016, 'Alain Farmer', 'alain.farmer@hotmail.it'], [3272, 'Renato Murillo', 'renato.murillo@club.com'], [1375, 'Edda Ferrari s', 'edda.ferraris@libero.it'], [4014, 'Luce Marty', 'luce.marty@waehner.org'], [3199, 'Rotraud Mitchell', 'rotraud.mitchell@laboratorios.com'], [2184, 'Israel Watkins', 'israel.watkins@hotmail.com'], [2170, 'Nathan Jennings', 'nathan.jennings@kim.com'], [1478, 'Gavin Proietti', 'gavin.proietti@gmx.de'], [6163, 'Emma Rios', 'emma.rios@yahoo.com.au'], [1889, 'Todd Douglas', 'todd.douglas@web.de'], [5912, 'Oscar H olsten', 'oscar.holsten@yahoo.com'], [2537, 'Ann Peron', 'ann.peron@grupo.biz'], [2118, 'Senta Grassi', 'senta.grassi@hotmail.com'], [3234, 'Kata Mena', 'kata.mena@gmail.com'], [3842, 'Gianfranco Hethur', 'g ianfranco.hethur@stewart-lee.com.au'], [3032, 'Florentine Saraceno', 'florentine.saraceno@hotmail.com'], [3238, 'Giuliano Bolzmann', 'giuliano.bolzmann@pacillo.it'], [2059, 'Alwine Kostolzin', 'alwine.kostolzi n@holloway.com'], [5238, 'Giacinto Briggs', 'giacinto.briggs@yahoo.co.uk'], [2578, 'Holly Negrete', 'holly.negrete@hotmail.com'], [4789, 'Mark Trapanese', 'mark.trapanese@gmail.com'], [2357, 'Alyssa Brun', 'alysa.brun@chittolini-spadafora.it'], [1235, 'Cristina Chevalier', 'cristina.chevalier@proyectos.com'], [3558, 'Shannon Lefevre', 'shannon.lefevre@gmail.com'], [1707, 'Logan Hornich', 'logan.hornich@tin.i t'], [2842, 'Rosalia Toro', 'rosalia.toro@free.fr'], [5318, 'Reiner Poerio', 'reiner.poerio@yahoo.com'], [1620, 'Valentina Pages', 'valentina.pages@hotmail.de'], [4998, 'Enrico Fabre', 'enrico.fabre@gmail.co m'], [3111, 'Sevim Begue', 'sevim.begue@yahoo.com.au'], [1424, 'Sonja Patberg', 'sonja.patberg@hotmail.com'], [3244, 'Christine Hiller', 'christine.hiller@yahoo.com'], [4643, 'Maik Beasley', 'maik.beasley@ha rper-brooks.biz'], [6132, 'Elliot Santana', 'elliott.santana@gmx.de'], [3790, 'Eloy Renzi', 'eloy.renzi@gmail.com'], [1312, 'Wolf Lutz', 'wolf.lutz@nichols-jackson.com.au'], [5686, 'Giuseppe Lee', 'giuseppe.l

ee@mahe.org'], [2111, 'Angelica Cuellar', 'angelica.cuellar@zaguri-bellucci.net'], [5685, 'Bekir Normand', 'bekir.normand@yahoo.de'], [1905, 'Carmine Gonzales', 'carmine.gonzales@schleich.net'], [3271, 'Marie Hubert', 'marianne.hubert@hotmail.com'], [4178, 'Gonzalo Montes', 'gonzalo.montes@hotmail.com'], [3275, 'Shelly Guichard', 'shelly.guichard@gmail.com'], [5713, 'Tobias Carabajal', 'tobias.carabajal@live.com'], [3339, 'Santiago Berry', 'santiago.berry@aol.de'], [5585, 'Emilio Davenport', 'emilio.davenport@zamora-russo.net'], [4487, 'Maurizio Gute', 'maurizio.gute@knight.info'], [1941, 'Nadia Travaglio', 'nadia.travaglio@colletti-broggini.com'], [4690, 'Elisa Barkholz', 'elisa.barkholz@sfr.fr'], [3607, 'Cristal Mazzini', 'cristal.mazzini@gmail.com'], [2670, 'Trevor Granados', 'trevor.granados@gmail.com'], [5261, 'Matthieu Mcdonald', 'matthieu.mcdonald@morrison.org'], [4121, 'Juliette Preston', 'juliette.preston@gmail.com'], [4620, 'Stanislaw Montalvo', 'stanislaw.montalvo@cmpherson-hughes.info'], [4630, 'Lilia Lollo brigida', 'lilia.lollobrigida@le.com'], [3440, 'Brigitta Shaw', 'brigitta.shaw@hotmail.co.uk'], [3572, 'Rebeca Frey', 'rebeca.frey@gerard.fr'], [5306, 'Nicolas Palacios', 'nicolas.palacios@meraz.com'], [5122, 'Susanna Bonnaeu', 'susanna.bonnaeu@klingelhoef.de'], [1368, 'Romina Valentine', 'romina.valentine@hotmail.com'], [5370, 'Erika Toussaint', 'erika.toussaint@sfr.fr'], [5636, 'Lothar Laurent', 'lothar.laurent@garcia.com.au'], [6033, 'Chelsea Verdugo', 'chelsea.verdugo@campbell.com'], [3599, 'Ria Venier', 'ria.venier@muriello-estevez.com'], [5566, 'Carly Summers', 'carly.summers@yahoo.co.uk'], [1873, 'Pasquale Tomasini', 'pasquale.tomasini@hotmail.co.uk'], [3904, 'Stefani Ughi', 'stefani.ughi@hotmail.it'], [4453, 'Jose Huang', 'jose.huang@gmx.de'], [5481, 'Petros Gaona', 'petros.gaona@googlemail.com'], [2987, 'Maximilian Siering', 'maximilian.siering@gaito-fornaciari.org'], [2431, 'Ermenegildo Holzapfel', 'ermenegildo.holzapfel@aol.de'], [4158, 'Damien Leger', 'damien.leger@dixon.biz'], [3753, 'Zeferino Barrera', 'zeferino.barrera@zedillo-velazquez.com'], [4391, 'Alfred Pohl', 'alfred.pohl@fechner.net'], [2026, 'Marissa Gaito', 'marissa.gaito@reed-campbell.edu'], [4547, 'Ariana Houston', 'ariana.houston@hotmail.it'], [2798, 'Victor Jacobs', 'victor.jacobs@young.com'], [4732, 'Henriette Lang', 'henriette.lang@mimun.it'], [5126, 'Wilfriede Baca', 'wilfriede.baca@caraballo.com'], [5151, 'Lucia Riviere', 'lucia.riviere@samaniego.biz'], [2134, 'Gordon Vigorelli', 'gordon.vigorelli@solano.com'], [5786, 'Diethard Salgado', 'diethard.salgado@gmail.com'], [5515, 'Sue Corrales', 'sue.corrales@cardenas-serna.org'], [4530, 'Arthur Salazar', 'arthur.salazar@yahoo.com.au'], [4088, 'Fortunata Schneider', 'fortunata.schneider@tiscali.it'], [3667, 'Gretchen Serna', 'gretchen.serna@pichon.org'], [3286, 'Egon Caron', 'egon.caron@hotmail.com'], [4804, 'Roberta Finzi', 'roberta.finzi@knight-davies.co.uk'], [5768, 'Editha Savage', 'editha.savage@davies.co.uk'], [6098, 'Allan Emanuelli', 'allan.emanuelli@fernandes.org'], [2508, 'Laurence Lucero', 'laurence.lucero@martin.org'], [5779, 'Istvan Draghi', 'istvan.draghi@france.com'], [1717, 'Luigi Bonnet', 'luigi.bonnet@carriera-federico.it'], [3368, 'Yuridia Fantozzi', 'yuridia.fantozzi@mcdonald.com'], [4265, 'Eraldo Schmiedt', 'eraldo.schmiedt@hotmail.de'], [4455, 'Antoinette Cox', 'antoinette.cox@bustamante.com'], [2940, 'Cameron Ocampo', 'cameron.ocampo@hotmail.com.au'], [6144, 'Lynn Lawson', 'lynn.lawson@elcercq.com'], [3657, 'Natalia Seifert', 'natalia.seifert@france.com'], [1537, 'Nuray Anguillara', 'nuray.anguillara@yahoo.co.uk'], [2116, 'Beate Morgan', 'beate.morgan@gmail.com'], [5102, 'Mario Rees', 'mario.rees@gmx.de'], [5815, 'Galasso Junken', 'galasso.junken@yahoo.co.uk'], [4250, 'Luca Moses', 'luca.moses@franke.de'], [4533, 'Aurelio Gertz', 'aurelio.gertz@hewitt.net'], [3329, 'Uberto Steele', 'uberto.steele@outlook.com'], [3606, 'Annie Hunter', 'annie.hunter@allen.co.uk'], [4749, 'Jo McCarty', 'jo.mccarty@hotmail.com'], [2222, 'Eitel Lombard', 'eitel.lombard@thomas.net'], [1939, 'Susan Vigliotti', 'susan.vigliotti@yahoo.co.uk'], [2456, 'Jacqueline Germano', 'jacqueline.germano@valette.fr'], [2794, 'Fidel Ruppert', 'fidel.ruppert@yahoo.co.uk'], [2648, 'Margaret Eigenwillig', 'margaret.eigenwillig@yahoo.com'], [2545, 'Yvette Harding', 'yvette.harding@yahoo.co.uk'], [3273, 'Federigo Karge', 'federigo.karge@bohlander.com'], [6155, 'Guenther Armenta', 'guenther.armenta@free.fr'], [6064, 'Coriolano Higgins', 'coriolano.higgins@murphy.org.au'], [1729, 'Felix Rubio', 'felix.rubio@ebert.de'], [4969, 'Melanie Leveque', 'melanie.leveque@fleury.net'], [3402, 'Abbie Galeati', 'abbie.galeati@ubaldi.it'], [2810, 'Vittorio Cacciopoli', 'vittorio.cacciopoli@yahoo.com'], [2217, 'Biagio Valenzuela', 'biagio.valenzuela@robinson.info'], [5140, 'Dario Green', 'dario.green@live.com'], [4497, 'Alwina Textor', 'alwina.textor@poste.it'], [5100, 'Diether Baeza', 'diether.baeza@grupo.biz'], [3390, 'Sole Trevino', 'sole.trevino@hotmail.com'], [1926, 'Darren Bernardi', 'darren.bernardi@brooks-martin.com'], [3264, 'Martina Hesser', 'martina.hesser@morrellato.it'], [2573, 'Bethan Giacometti', 'bethan.giacometti@gmail.com'], [3628, 'Ethan Duhamel', 'ethan.duhamel@sfr.fr'], [3826, 'Corrado Warren', 'corrado.warren@hotmail.com.au'], [5983, 'Damian Pulido', 'damian.pulido@yahoo.de'], [5678, 'Dino Osborne', 'dino.osborne@despacho.com'], [2093, 'Nicolaus Bernetti', 'nicolaus.bernetti@aol.de'], [5333, 'Gianna Carroll', 'gianna.carroll@perez.com'], [5139, 'Grit Knight', 'grit.knight@web.de'], [3173, 'Erin Weinhold', 'erin.weinhold@hotmail.fr'], [3333, 'Craig Jordan', 'craig.jordan@gmail.com'], [2262, 'Homero Respighi', 'homero.respighi@gmail.com'], [1531, 'Fanny Pertile', 'fanny.pertile@gmail.com'], [4881, 'Fridolin Comboni', 'fridolin.comboni@graf.com'], [3928, 'Hugh Cantu', 'hugh.cantu@libero.it'], [1803, 'Alphons Simmons', 'alphons.simmons@hotmail.com'], [2321, 'Ludovico Cafarchia', 'ludovico.cafarchia@gmail.com'], [6234, 'Nico Kaiser', 'nico.kaiser@almonete.com'], [1422, 'Kirsty Lamy', 'kirsty.lamy@alonzii.it'], [2660, 'Joshua Scheibe', 'joshua.scheibe@turci.org'], [5361, 'Carin Pineau', 'carin.pineau@yahoo.com'], [3420, 'Ellie Galarza', 'ellie.galarza@pareto-cattaneo.net'], [4337, 'Marion Lira', 'marion.lira@ifrance.com'], [2338, 'Aurelia Angioiello', 'aurelia.angioiello@lorid.com'], [4140, 'Maura Ravaglioli', 'maura.ravaglioli@yahoo.com'], [2373, 'Brunhilde Fletcher', 'brunhilde.fletcher@schomber.de'], [1344, 'Ottofried Marques', 'ottofried.marques@tyler-watson.info'], [2141, 'Christelle Blasi', 'christelle.blasi@morris-chan.com'], [3415, 'Augusto Vercelloni', 'augusto.vercelloni@club-internet.fr'], [5696, 'Michela Pruvost', 'michela.pruvost@holt.co.uk'], [1798, 'Guarino Escalante', 'guarino.escalante@hotmail.com'], [1657, 'Gabriela Padilla', 'gabriela.padilla@pearson-russell.com'], [3651, 'Agathe Dias', 'agathe.dias@jackson.com'], [3574, 'Werner Kallert', 'werner.kallert@gmail.com'], [6063, 'Lore Delaunay', 'lore.delaunay@yahoo.com'], [2404, 'Enzio Campos', 'enzio.campos@tim.it'], [2705, 'Walther Duarte', 'walther.duarte@web.de'], [5059, 'Adelgunde Trevisan', 'adelgunde.trevisan@gmail.com'], [5466, 'Thierry Schwartz', 'thierry.schwartz@travaglio.net'], [5821, 'Katie Andersen', 'katie.andersen@haney.com.au'], [5385, 'Kathy Zorani', 'kathy.zorani@hotmail.com'], [3506, 'Giuseppina Tanguy', 'giuseppina.tanguy@wanadoo.fr'], [5427, 'Jack Klapp', 'jack.klapp@hotmail.com.au'], [6142, 'Tom Leconte', 'tom.leconte@woods-johnson.com'], [2873, 'Bonnie Cignaroli', 'bonnie.cignaroli@yahoo.com'], [3916, 'Dannan Farrell', 'dannan.farrell@hotmail.com'], [1687, 'Fiorenzo Bauer', 'fiorenzo.bauer@gmail.com'], [1542, 'Randy Alfieri', 'randy.alfieri@verri.org'], [3087, 'Tim Bradley', 'tim.bradley@schlosser.de'], [3857, 'Yves Mendez', 'yves.mendez@gmail.com'], [5921, 'Raffaella Esquivel', 'raffaella.esquivel@gmail.com'], [3144, 'Lawrence Vasari', 'lawrence.vasari@castillo.edu'], [2676, 'Sam Adams', 'sam.adams@yahoo.com'], [3646, 'Mirco Nicolini', 'mirco.nicolini@blin.net'], [1321, 'Graham Bajardi', 'graham.bajardi@yahoo.com'], [1399, 'Luc Quinn', 'luc.quinn@walker.com'], [3169, 'Stella Ullrich', 'stella.ullrich@hotmail.com.au'], [2058, 'Calcedonio Remy', 'calcedonio.remy@edwards-benton.com'], [1300, 'Nanni Almaraz', 'nanni.almaraz@martin.com'], [2075, 'Pascual Corradi', 'pascual.corradi@lopez.net.au'], [2984, 'Kristi Grant', 'kristi.grant@yahoo.co.uk'], [4768, 'Manuel Gracia', 'manuel.gracia@trussardi.eu'], [3682, 'Gionata Hussain', 'gionata.hussain@yahoo.com'], [5901, 'Maria Nguyen', 'maria.nguyen@hernandez.com'], [3540, 'Leon Baker', 'leon.baker@hotmail.com'], [2225, 'Gianmarco Alcaraz', 'gianmarco.alcaraz@zaccchia.net'], [1485, 'Derrick Little', 'derrick.little@small.org'], [5343, 'Terry Ramirez', 'terry.ramirez@hotmail.co.uk'], [5594, 'Katarina Clarke', 'katarina.clarke@hotmail.com.au'], [2482, 'Leila Washington', 'leila.washington@gmail.com'], [2706, 'Mirko Townsend', 'mirko.townsend@schmitt.com'], [4861, 'Randall Connolly', 'randall.connolly@jovinelli-castelli.com'], [2961, 'Pia Burgess', 'pia.burgess@proyectos.org'], [3043, 'Justine Mancini', 'justine.mancini@tiscali.it'], [6103, 'Reynaldo Wagenknecht', 'reynaldo.wagenknecht@faure.com'], [4708, 'Toni Magrassi', 'toni.magrassi@gmail.com'], [4395, 'Patrizio Cisneros', 'patrizio.cisneros@martinez.edu.au'], [4912, 'Durante Lindner', 'durante.lindner@picard.fr'], [2030, 'Rosli Lamborghini', 'rosli.lamborghini@hiller.net'], [5235, 'Ugo Harrison', 'ugo.harrison@morley-chapman.co.uk'], [5001, 'Isidor Carpenter', 'isidor.carpenter@chauvin.com'], [4093, 'Marie Zoppetto', 'marie.zoppetto@kallert.de'], [2576, 'Branko Chambers', 'branko.chambers@ifrance.com'], [6122, 'Alan Pineda', 'alan.pineda@proyectos.info'], [2813, 'Carmen Oestrovsky', 'carmen.oestrovsky@gmail.com'], [3737, 'Shane Fernandez', 'shane.fernandez@hotmail.de'], [2803, 'Debbie Guilbert', 'debbie.guilbert@berg.edu'], [5508, 'Vito Dossi', 'vito.dossi@hotmail.co.uk'], [3850, 'Evan Schacht', 'evan.schacht@price.com'], [4220, 'Mandy Weston', 'mandy.weston@free.fr'], [4532, 'Paulo Zito', 'paulo.zito@nelson.biz'], [6013, 'Alberto Girschner', 'alberto.girschner@laboratorios.com'], [3396, 'Giacomo Bouvet', 'giacomo.bouvet@gmail.com'], [4401, 'Ehrentraud Puga', 'ehrentraud.puga@laboratorios.com'], [2981, 'Cristian Hale', 'cristian.hale@hotmail.co.uk'], [4467, 'Lilija Charles', 'lilija.charles@gmail.com'], [1377, 'Albert Legendre', 'albert.legendre@lee.com'], [5036, 'Marisa Renaud', 'marisa.renaud@smith.org'], [5761, 'Sharon Dially', 'sharon.dially@gmail.com'], [3699, 'Eduardo Delattre', 'eduardo.delattre@lynch.net.au'], [4519, 'Bernadette Wang', 'bernadette.wang@gmail.com'], [5395, 'Dina Gibbons', 'dina.gibbons@brewer.net'], [3057, 'Clara Armstrong', 'clara.armstrong@vodafone.it'], [2741, 'Elliott Herrmann', 'elliott.herrmann@hotmail.com.au'], [1675, 'Graziela Vidal', 'graziela.vidal@hotmail.com'], [2712, 'Francisca Becerra', 'francisca.becerra@young.com.au'], [4527, 'Edeltrud Boyer', 'edeltrud.boyer@outlook.com'], [1722, 'Julian Noble', 'julian.noble@hill.org.au'], [4330, 'Christophe Bohnbach', 'christophe.bohnbach@ifrance.com'], [5219, 'Ippazio Angulo', 'ippazio.angulo@yahoo.com'], [4997, 'Grace Kroker', 'grace.kroker@hotmail.com'], [2239, 'Jared Ferguson', 'jared.ferguson@roberts.com'], [4587, 'Virginie McLean', 'virginie.mclean@yahoo.com'], [4113, 'Mason Potter', 'mason.potter@gmail.com'], [4762, 'Aaron Anaya', 'aaron.anaya@hotmail.com.au'], [5242, 'Leah Ornelas', 'leah.ornelas@gmail.com'], [5765, 'Harald Roy', 'harald.roy@web.de'], [2185, 'Micaela Martel', 'micaela.martel@hotmail.com'], [5500, 'Vincenzo Sanders', 'vincenzo.sanders@voila.fr'], [3494, 'Bettina Norbiato', 'bettina.norbiato@allen-lutz.org'], [3924, 'Julio Renner', 'julio.renner@industrias.net'], [3521, 'Jelena Stiffel', 'jelena.stiffel@trupp.de'], [5262, 'Guadalupe Munoz', 'guadalupe.munoz@murray-hamilton.com.au'], [3684, 'Whitney Noack', 'whitney.noack@laboratorios.org'], [5784, 'Gelsomina Migliaccio', 'gelsomina.migliaccio@junk.com'], [1498, 'Evangelista Pereira', 'evangelista.pereira@thompson-peterson.biz'], [6073, 'Gareth Comolli', 'gareth.comolli@tiscali.fr'], [4939, 'Michelangelo Hess', 'michelangelo.hess@bouygtel.fr']]

```
In [ ]: # Create a contact_info DataFrame -
# 1. Create two lists, one for the keys (column_names) and one for the values
values = []
column_names = []

# 2. Iterate through the DataFrame
for i, row in contact_info_df_orig.iterrows():
    data = row.iloc[0]
```

```

# Convert each row to a Python dictionary.
converted_data = json.loads(data)

# Use a List comprehension to get the keys from the converted data.
columns = [k for k, v in converted_data.items()]

# Use a List comprehension to get the values for each row.
row_values = [v for k, v in converted_data.items()]

# Append the keys and list values to the lists created in step 1.
column_names.append(columns)
values.append(row_values)

contact_info_df = pd.DataFrame(values, columns=column_names[0])
contact_info_df.head()

```

```
Out[ ]:
```

| | contact_id | name | email |
|---|------------|-------------------|------------------------------|
| 0 | 4661 | Cecilia Velasco | cecilia.velasco@rodrigues.fr |
| 1 | 3765 | Mariana Ellis | mariana.ellis@rossi.org |
| 2 | 4187 | Sofie Woods | sofie.woods@riviere.com |
| 3 | 4941 | Jeanette Iannotti | jeanette.iannotti@yahoo.com |
| 4 | 2199 | Samuel Sorgatz | samuel.sorgatz@gmail.com |

```

In [ ]: # Check the datatypes.
contact_info_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   contact_id  1000 non-null    int64
1   name        1000 non-null    object
2   email       1000 non-null    object
dtypes: int64(1), object(2)
memory usage: 23.6+ KB

```

```

In [ ]: print(contact_info_df.columns)

Index(['contact_id', 'name', 'email'], dtype='object')

```

```

In [ ]: # Split the 'name' column into 'first_name' and 'last_name'
contact_info_df[['first_name', 'last_name']] = contact_info_df['name'].str.split(' ', expand=True)

# Drop the 'name' column
contact_info_df=contact_info_df.drop(columns=['name'])

contact_info_df.head()

```

```
Out[ ]:
```

| | contact_id | email | first_name | last_name |
|---|------------|------------------------------|------------|-----------|
| 0 | 4661 | cecilia.velasco@rodrigues.fr | Cecilia | Velasco |
| 1 | 3765 | mariana.ellis@rossi.org | Mariana | Ellis |
| 2 | 4187 | sofie.woods@riviere.com | Sofie | Woods |
| 3 | 4941 | jeanette.iannotti@yahoo.com | Jeanette | Iannotti |
| 4 | 2199 | samuel.sorgatz@gmail.com | Samuel | Sorgatz |

```

In [ ]: # Reorder the columns
contact_info_df = contact_info_df[['contact_id', 'first_name', 'last_name', 'email']]
contact_info_df.head()

```



```
Out[ ]:
```

| | contact_id | first_name | last_name | email |
|---|------------|------------|-----------|------------------------------|
| 0 | 4661 | Cecilia | Velasco | cecilia.velasco@rodrigues.fr |
| 1 | 3765 | Mariana | Ellis | mariana.ellis@rossi.org |
| 2 | 4187 | Sofie | Woods | sofie.woods@riviere.com |
| 3 | 4941 | Jeanette | Iannotti | jeanette.iannotti@yahoo.com |
| 4 | 2199 | Samuel | Sorgatz | samuel.sorgatz@gmail.com |

```
In [ ]: # Check the datatypes one more time before exporting as CSV file.
contact_info_df.dtypes
```

```
Out[ ]:
```

| | |
|------------|--------|
| contact_id | int64 |
| first_name | object |
| last_name | object |
| email | object |
| dtype: | object |

```
In [ ]: # Export the DataFrame as a CSV file.
contact_info_df.to_csv("CSV files/contacts.csv", encoding='utf8', index=False)
```

Option 2: Use regex to create the contacts DataFrame.

```
In [ ]: contact_info_df_copy = contact_info_df_orig.copy()
contact_info_df_copy.head()
```

```
Out[ ]:
```

| | contact_info |
|---|---|
| 0 | {"contact_id": 4661, "name": "Cecilia Velasco", "email": "cecilia.velasco@rodrigues.fr"} |
| 1 | {"contact_id": 3765, "name": "Mariana Ellis", "email": "mariana.ellis@rossi.org"} |
| 2 | {"contact_id": 4187, "name": "Sofie Woods", "email": "sofie.woods@riviere.com"} |
| 3 | {"contact_id": 4941, "name": "Jeanette Iannotti", "email": "jeanette.iannotti@yahoo.com"} |
| 4 | {"contact_id": 2199, "name": "Samuel Sorgatz", "email": "samuel.sorgatz@gmail.com"} |

```
In [ ]: # Extract the four-digit contact ID number.
contact_info_df_copy['contact_id'] = contact_info_df_copy['contact_info'].str.extract(r'(\d+)')
contact_info_df_copy.head()
```

```
Out[ ]:
```

| | contact_info | contact_id | name | email |
|---|---|------------|-------------------|------------------------------|
| 0 | {"contact_id": 4661, "name": "Cecilia Velasco", "email": "cecilia.velasco@rodrigues.fr"} | 4661 | Cecilia Velasco | cecilia.velasco@rodrigues.fr |
| 1 | {"contact_id": 3765, "name": "Mariana Ellis", "email": "mariana.ellis@rossi.org"} | 3765 | Mariana Ellis | mariana.ellis@rossi.org |
| 2 | {"contact_id": 4187, "name": "Sofie Woods", "email": "sofie.woods@riviere.com"} | 4187 | Sofie Woods | sofie.woods@riviere.com |
| 3 | {"contact_id": 4941, "name": "Jeanette Iannotti", "email": "jeanette.iannotti@yahoo.com"} | 4941 | Jeanette Iannotti | jeanette.iannotti@yahoo.com |
| 4 | {"contact_id": 2199, "name": "Samuel Sorgatz", "email": "samuel.sorgatz@gmail.com"} | 2199 | Samuel Sorgatz | samuel.sorgatz@gmail.com |

```
In [ ]: # Check the datatypes.
contact_info_df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   contact_info 1000 non-null   object
1   contact_id   1000 non-null   object
dtypes: object(2)
memory usage: 15.8+ KB
```

```
In [ ]: # Convert the "contact_id" column to an int64 data type.
contact_info_df_copy['contact_id'] = pd.to_numeric(contact_info_df_copy['contact_id'])
contact_info_df_copy.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   contact_info 1000 non-null   object
1   contact_id   1000 non-null   int64
dtypes: int64(1), object(1)
memory usage: 15.8+ KB

In [ ]: # Extract the name of the contact and add it to a new column.
contact_info_df_copy['name'] = contact_info_df_copy['contact_info'].str.extract(r'"name":\s*"([^\"]+)"')
contact_info_df_copy.head(10)
```

Out[]:

| | contact_info | contact_id | name |
|---|---|------------|-------------------|
| 0 | {"contact_id": 4661, "name": "Cecilia Velasco", "email": "cecilia.velasco@rodrigues.fr"} | 4661 | Cecilia Velasco |
| 1 | {"contact_id": 3765, "name": "Mariana Ellis", "email": "mariana.ellis@rossi.org"} | 3765 | Mariana Ellis |
| 2 | {"contact_id": 4187, "name": "Sofie Woods", "email": "sofie.woods@riviere.com"} | 4187 | Sofie Woods |
| 3 | {"contact_id": 4941, "name": "Jeanette Iannotti", "email": "jeanette.iannotti@yahoo.com"} | 4941 | Jeanette Iannotti |
| 4 | {"contact_id": 2199, "name": "Samuel Sorgatz", "email": "samuel.sorgatz@gmail.com"} | 2199 | Samuel Sorgatz |
| 5 | {"contact_id": 5650, "name": "Socorro Luna", "email": "socorro.luna@hotmail.com"} | 5650 | Socorro Luna |
| 6 | {"contact_id": 5889, "name": "Carolina Murray", "email": "carolina.murray@knight.com"} | 5889 | Carolina Murray |
| 7 | {"contact_id": 4842, "name": "Kayla Moon", "email": "kayla.moon@yahoo.de"} | 4842 | Kayla Moon |
| 8 | {"contact_id": 3280, "name": "Ariadna Geisel", "email": "ariadna.geisel@rangel.com"} | 3280 | Ariadna Geisel |
| 9 | {"contact_id": 5468, "name": "Danielle Ladeck", "email": "danielle.ladeck@scalfaro.net"} | 5468 | Danielle Ladeck |

```
In [ ]: # Extract the email from the contacts and add the values to a new column.
contact_info_df_copy['email'] = contact_info_df_copy['contact_info'].str.extract(r'"email":\s*"([^\"]+)"')
contact_info_df_copy.head(10)
```

Out[]:

| | contact_info | contact_id | name | email |
|---|---|------------|-------------------|------------------------------|
| 0 | {"contact_id": 4661, "name": "Cecilia Velasco", "email": "cecilia.velasco@rodrigues.fr"} | 4661 | Cecilia Velasco | cecilia.velasco@rodrigues.fr |
| 1 | {"contact_id": 3765, "name": "Mariana Ellis", "email": "mariana.ellis@rossi.org"} | 3765 | Mariana Ellis | mariana.ellis@rossi.org |
| 2 | {"contact_id": 4187, "name": "Sofie Woods", "email": "sofie.woods@riviere.com"} | 4187 | Sofie Woods | sofie.woods@riviere.com |
| 3 | {"contact_id": 4941, "name": "Jeanette Iannotti", "email": "jeanette.iannotti@yahoo.com"} | 4941 | Jeanette Iannotti | jeanette.iannotti@yahoo.com |
| 4 | {"contact_id": 2199, "name": "Samuel Sorgatz", "email": "samuel.sorgatz@gmail.com"} | 2199 | Samuel Sorgatz | samuel.sorgatz@gmail.com |
| 5 | {"contact_id": 5650, "name": "Socorro Luna", "email": "socorro.luna@hotmail.com"} | 5650 | Socorro Luna | socorro.luna@hotmail.com |
| 6 | {"contact_id": 5889, "name": "Carolina Murray", "email": "carolina.murray@knight.com"} | 5889 | Carolina Murray | carolina.murray@knight.com |
| 7 | {"contact_id": 4842, "name": "Kayla Moon", "email": "kayla.moon@yahoo.de"} | 4842 | Kayla Moon | kayla.moon@yahoo.de |
| 8 | {"contact_id": 3280, "name": "Ariadna Geisel", "email": "ariadna.geisel@rangel.com"} | 3280 | Ariadna Geisel | ariadna.geisel@rangel.com |
| 9 | {"contact_id": 5468, "name": "Danielle Ladeck", "email": "danielle.ladeck@scalfaro.net"} | 5468 | Danielle Ladeck | danielle.ladeck@scalfaro.net |

```
In [ ]: # Create a copy of the contact_info_df with the 'contact_id', 'name', 'email' columns.
new_contact_info_df = contact_info_df_copy[['contact_id', 'name', 'email']].copy()
new_contact_info_df.head(10)
```

```
Out[ ]:
```

| | contact_id | name | email |
|---|------------|-------------------|------------------------------|
| 0 | 4661 | Cecilia Velasco | cecilia.velasco@rodrigues.fr |
| 1 | 3765 | Mariana Ellis | mariana.ellis@rossi.org |
| 2 | 4187 | Sofie Woods | sofie.woods@riviere.com |
| 3 | 4941 | Jeanette Iannotti | jeanette.iannotti@yahoo.com |
| 4 | 2199 | Samuel Sorgatz | samuel.sorgatz@gmail.com |
| 5 | 5650 | Socorro Luna | socorro.luna@hotmail.com |
| 6 | 5889 | Carolina Murray | carolina.murray@knight.com |
| 7 | 4842 | Kayla Moon | kayla.moon@yahoo.de |
| 8 | 3280 | Ariadna Geisel | ariadna.geisel@rangel.com |
| 9 | 5468 | Danielle Ladeck | danielle.ladeck@scalfaro.net |

```
In [ ]: # Create a "first_name" and "last_name" column with the first and last names from the "name" column.
new_contact_info_df[['first_name', 'last_name']] = new_contact_info_df['name'].str.split(' ', n=1, expand=True)

# Drop the contact_name column
contact_info_clean = new_contact_info_df.drop(columns=['name'])
contact_info_clean.head(10)
```

```
Out[ ]:
```

| | contact_id | email | first_name | last_name |
|---|------------|------------------------------|------------|-----------|
| 0 | 4661 | cecilia.velasco@rodrigues.fr | Cecilia | Velasco |
| 1 | 3765 | mariana.ellis@rossi.org | Mariana | Ellis |
| 2 | 4187 | sofie.woods@riviere.com | Sofie | Woods |
| 3 | 4941 | jeanette.iannotti@yahoo.com | Jeanette | Iannotti |
| 4 | 2199 | samuel.sorgatz@gmail.com | Samuel | Sorgatz |
| 5 | 5650 | socorro.luna@hotmail.com | Socorro | Luna |
| 6 | 5889 | carolina.murray@knight.com | Carolina | Murray |
| 7 | 4842 | kayla.moon@yahoo.de | Kayla | Moon |
| 8 | 3280 | ariadna.geisel@rangel.com | Ariadna | Geisel |
| 9 | 5468 | danielle.ladeck@scalfaro.net | Danielle | Ladeck |

```
In [ ]: # Reorder the columns
contact_info_clean = contact_info_clean[['contact_id', 'first_name', 'last_name', 'email']]
contact_info_clean.head(10)
```

```
Out[ ]:
```

| | contact_id | first_name | last_name | email |
|---|------------|------------|-----------|------------------------------|
| 0 | 4661 | Cecilia | Velasco | cecilia.velasco@rodrigues.fr |
| 1 | 3765 | Mariana | Ellis | mariana.ellis@rossi.org |
| 2 | 4187 | Sofie | Woods | sofie.woods@riviere.com |
| 3 | 4941 | Jeanette | Iannotti | jeanette.iannotti@yahoo.com |
| 4 | 2199 | Samuel | Sorgatz | samuel.sorgatz@gmail.com |
| 5 | 5650 | Socorro | Luna | socorro.luna@hotmail.com |
| 6 | 5889 | Carolina | Murray | carolina.murray@knight.com |
| 7 | 4842 | Kayla | Moon | kayla.moon@yahoo.de |
| 8 | 3280 | Ariadna | Geisel | ariadna.geisel@rangel.com |
| 9 | 5468 | Danielle | Ladeck | danielle.ladeck@scalfaro.net |

```
In [ ]: # Check the datatypes one more time before exporting as CSV file.
contact_info_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    contact_id  1000 non-null   object
1    first_name  1000 non-null   object
2    last_name   1000 non-null   object
3    email       1000 non-null   object
dtypes: object(4)
memory usage: 31.4+ KB
```

```
In [ ]: # Export the DataFrame as a CSV file.
contact_info_clean.to_csv("Resources/contacts_regex.csv", encoding='utf8', index=False)
```

```
In [ ]:
```