In [2]:
```
import os
import pandas as pd
pd.options.display.float_format = '{:20,.2f}'.format
pd.set_option('display.max_rows', 5000)
pd.set_option('display.max_columns', 5000)
pd.set_option('display.width', 1000)
pd.set_option('display.max_colwidth', -1)
```

In [3]:
```
amazon_google_path = (r'/home/ubuntu/jupyter/ServerX/1_Standard Data In
                      r'/Unprocessed Data/product_samples/amazon_google
```

Amazon dataset

In [4]:
```
az_fields = ['id', 'title', 'description', 'manufacturer', 'price']
```

In [8]:
```
amazon_data = pd.read_csv(
    os.path.join(amazon_google_path, 'Amazon.csv'),
    sep = ',',
    quotechar = '"',
    encoding = 'latin-1',
    usecols  = az_fields,
)[az_fields]
```

In [9]:
```
amazon_data.rename(
    columns = {
        'title': 'name',
        'description': 'prod_descr',
        'manufacturer': 'producer',
    },
    inplace = True
)
```

In [11]:
```
amazon_data.head(2)
```

Out[11]:

	id	name	prod_descr	producer	price
0	b000jz4hqo	clickart 950 000 - premier image pack (dvd-rom)	NaN	broderbund	0.00
1	b0006zf55o	ca international - arcserve lap/desktop oem 30pk	oem arcserve backup v11.1 win 30u for laptops and desktops	computer associates	0.00

Google Dataset

In [13]:
```
google_fields = ['id', 'name', 'description', 'manufacturer', 'price']
```

In [14]:
```
google_data = pd.read_csv(
    os.path.join(amazon_google_path, 'GoogleProducts.csv'),
    sep = ',',
    quotechar = '"',
    encoding = 'latin-1',
    usecols  = google_fields,
)[google_fields]
```

In [15]:
```
google_data.rename(
    columns = {
        'description': 'prod_descr',
        'manufacturer': 'producer'
    },
    inplace = True
)
```

In [17]:
```
google_data.head(3)
```

Out[17]:

	id	name	prod_descr	produce
0	http://www.google.com/base/feeds/snippets/11125907881740407428	learning quickbooks 2007	learning quickbooks 2007	intu
1	http://www.google.com/base/feeds/snippets/11538923464407758599	superstart! fun with reading & writing!	fun with reading & writing! is designed to help kids learn to read and write better through exercises puzzle-solving creative writing decoding and more!	NaN
2	http://www.google.com/base/feeds/snippets/11343515411965421256	qb pos 6.0 basic software	qb pos 6.0 basic retail mngmt software. for retailers who need basic inventory sales and customer tracking.	intu

Mapping of Entries

In [18]:
```
map_az_go_data = pd.read_csv(
    os.path.join(amazon_google_path, 'Amzon_GoogleProducts_perfectMappi
    sep = ',',
    quotechar = '"',
    encoding = 'latin-1',
    #usecols  = google_fields,
)#[google_fields]
```

In [19]:
```
map_az_go_data.rename(
    columns = {
        'idAmazon': 'amazon_id',
        'idGoogleBase': 'google_id'
    },
    inplace=True
)
```

In [20]:
```
map_az_go_data.head()
```

Out[20]:

	amazon_id	google_id
0	b000jz4hqo	http://www.google.com/base/feeds/snippets/18441480711193821750
1	b00004tkvy	http://www.google.com/base/feeds/snippets/18441110047404795849
2	b000g80lqo	http://www.google.com/base/feeds/snippets/18441188461196475272
3	b0006se5bq	http://www.google.com/base/feeds/snippets/18428750969726461849
4	b00021xhzw	http://www.google.com/base/feeds/snippets/18430621475529168165

Artificial concatenation of Amazon + Google set

In [21]:
```
amazon_data['source'] = 'amazon'
google_data['source'] = 'google'
```

In [22]:
```
leipzig_product_data = pd.concat([amazon_data, google_data])
```

In [24]:
```
products_all_path = (r'/home/ubuntu/jupyter/ServerX/1_Standard Data Int
                     r'/Processed Data/product_samples/')
```

In [25]:
```
leipzig_product_data.to_csv(products_all_path + 'leipzig_product_data_a
```

In [26]:
```
map_az_go_data.to_csv(products_all_path + 'leipzig_product_mapping_data
```