Предварительная обработка данных

Импортируем необходимые библиотеки. на данном этапе достачно библиотеки pandas

In [136]:

```
import pandas
```

Исследуем файлы для подготовки их к формированию целевой базы данных

Подключаем файл customers_data.xls для анализа

In [137]:

```
df_customers_data = pandas.read_excel("a/xls/customers_data.xls")
df_customers_data.info()
```

```
<class 'pandas.DataFrame'>
```

```
RangeIndex: 4977 entries, 0 to 4976
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Customer ID           4977 non-null   str
 1   Customer Name         4977 non-null   str
 2   Customer Payment Terms  4977 non-null   str
 3   Address               4977 non-null   str
 4   Credit Limit          4977 non-null   int64
dtypes: int64(1), str(4)
memory usage: 194.5 KB
```

Анализируем данные в датафрейме df_customers_data

In [138]:

```
df_customers_data
```

Out[138]:

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
0	C0200769623-0	WAL-MAR corp	NAH4	55599 Katherine Harbors Suite 551\nWest Brenda...	50000
1	C02009808	BEN E	NAD1	5488 Michael	50000

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
	28-1			Inlet\nElizabethport, MP 17624	
2	C02007927 34-2	MDV/ trust	NAA8	708 Taylor Cape\nJohnstad, MT 34743	100000
3	C01401056 86-3	SYSC llc	CA10	4113 Dana Ridges\nEast Clarencestad, IA 61466	100000
4	C01401061 81-4	WAL-MAR foundation	NAH4	2759 Kimberly Villages\nThompsonside, OR 79370	100000
...	...	...	...	...	...
4972	C75386549 198-4995	Hull Inc	NATM	05134 Katherine Springs Suite 090\nJonathanvie...	20000
4973	C28188290 658-4996	Ramirez-Cain	NATH	34213 Jennings Land Suite 112\nMaxwellchester,...	1000000

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
4974	C20595783803-4997	Sims PLC	NATU	91232 Brown Valleys\nPort Austin, MD 81384	20000
4975	C84713226861-4998	Ferrell-Thomas	NATH	PSC 6545, Box 4898\nAPO AE 95400	100000
4976	C32572229109-4999	Alvarez-Floyd	NAC6	69237 Smith Passage\nJaredfort, AS 91937	100000

4977 rows × 5 columns

Вывод: В датафрейме содержатся ID покупателя, имя клиента, условия оплаты, адрес и кредитный лимит. Из данных, представленных в датафрейме, можно предположить, что в данном датафрейме содержится данные связанные с дебиторской задолженностью.

Подсчитываем пустые значения по всем столбцам датафрейма df_customers_data

In [139]:

```
df_customers_data.isna().sum()
```

Out[139]:

```
Customer ID              0
Customer Name            0
Customer Payment Terms   0
Address                  0
Credit Limit             0
dtype: int64
```

Подсчитываем количество дубликатов в датафрейме df_customers_data

In [140]:

```
df_customers_data.duplicated().sum()
```

Out[140]:

```
np.int64(0)
```

Исследуем выбросы

Метод describe() в библиотеке Pandas генерирует описательные статистики для столбцов DataFrame. Он вычисляет различные статистические показатели, такие как

количество записей, среднее значение, стандартное отклонение, минимум, максимум и перцентили для числовых столбцов. Также метод предоставляет обобщённые статистические данные для столбцов с типом данных объекта

In [141]:

```
df_customers_data.describe()
```

Out[141]:

	Credit Limit
count	4977.000000
mean	82194.092827
std	172857.225127
min	0.000000
25%	10000.000000
50%	20000.000000
75%	100000.000000
max	1000000.000000

In [142]:

```
df_customers_data['Credit Limit'].describe()
```

Out[142]:

```
count       4977.000000
mean       82194.092827
std       172857.225127
min            0.000000
25%        10000.000000
50%        20000.000000
75%       100000.000000
max      1000000.000000
Name: Credit Limit, dtype: float64
```

Вывод: Присутствует странный разброс значений в столбце 'Credit Limit', например минимальное нулевое значение. Кредитный лимит не может равняться нулю

Выведем нулевые значения

In [143]:

```
df_customers_data[df_customers_data['Credit Limit'] == 0]
```

Out[143]:

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
94	C0100036060-94	BIRD-I	NA10	1165 Robert Estate\nCarolshire, ME 37387	0
158	C0200793568-158	NICH corp	NAA8	Unit 7163 Box 8239\nDPO AP 27060	0
253	C0100016691-253	JAVA	NAA8	86645 Jennifer Mall\nMorganfort, IN 12809	0
263	C0140104475-263	FOOD trust	NAA8	69288 Cole Plains Suite 776\nNew Marcus, KS 65287	0
268	C0100005765-268	HT HA foundation	NAA8	32322 Durham Hill\nPort Elizabethtown, VA 14770	0
...	...	...	...	...	...
4179	C97244360592-4179	PLAZA WA associates	NAG2	81540 Tammy Road Suite 296\nDuranbury, IL	0

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
				94233	
4180	C72506571732-4180	BARB associates	NAA8	196 Gary Street Apt. 917\nNew Joshuafort, PW 7...	0
4181	C37008718163-4181	PLAZA WA llc	NAG2	206 Gregory Mountain\n Maloneport, OR 56931	0
4183	C67943642515-4183	JALI	NAA8	106 Brown Curve Suite 797\nLawrenceville, ID 0...	0
4189	C19808856070-4189	SOLOM associates	NAVE	890 Meredith Inlet Suite 110\nJenniferville, H...	0

403 rows × 5 columns

Как видно в датафрейме присутствуют 403 строки содержащие ноль в столбце 'Credit Limit'. Удалим их

In [144]:

```
df_customers_data = df_customers_data[df_customers_data['Credit Limit'] != 0]
```

Проверяем

In [145]:

```
df_customers_data
```

Out[145]:

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
0	C0200769623-0	WAL-MAR corp	NAH4	55599 Katherine Harbors Suite 551\nWest Brenda...	50000
1	C0200980828-1	BEN E	NAD1	5488 Michael Inlet\nElizabethport, MP 17624	50000
2	C02007927	MDV/ trust	NAA8	708 Taylor Cape\nJoh	100000

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
	34-2			nstad, MT 34743	
3	C01401056 86-3	SYSC llc	CA10	4113 Dana Ridges\nEast Clarencestad, IA 61466	100000
4	C01401061 81-4	WAL-MAR foundation	NAH4	2759 Kimberly Villages\nThompsonside, OR 79370	100000
...	...	...	...	...	...
4972	C75386549 198-4995	Hull Inc	NATM	05134 Katherine Springs Suite 090\nJonathanvie...	20000
4973	C28188290 658-4996	Ramirez-Cain	NATH	34213 Jennings Land Suite 112\nMaxwellchester,...	1000000
4974	C20595783 803-4997	Sims PLC	NATU	91232 Brown Valleys\nPort Austin, MD 81384	20000

	Customer ID	Customer Name	Customer Payment Terms	Address	Credit Limit
4975	C84713226861-4998	Ferrell-Thomas	NATH	PSC 6545, Box 4898\nAPO AE 95400	100000
4976	C32572229109-4999	Alvarez-Floyd	NAC6	69237 Smith Passage\nJaredfort, AS 91937	100000

4574 rows × 5 columns

In [146]:

```
df_customers_data['Credit Limit'].describe()
```

Out[146]:

```
count       4574.000000
mean       89435.942282
std       178507.610874
min         5000.000000
25%        20000.000000
50%        50000.000000
75%       100000.000000
max      1000000.000000
Name: Credit Limit, dtype: float64
```

Вывод: Как видно теперь минимальное значение по столбцу 'Credit Limit' составляет

5000 и общее количество записей в датафрейме составляет 4574

Подключаем файл payables_data.xls для анализа

In [147]:

```
df_payables_data = pandas.read_excel("a/xls/payables_data.xls")
df_payables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Invoice Number                        20000 non-null  str
 1   Posting Date                          20000 non-null  object
 2   Invoice Date                          20000 non-null  object
 3   Payment Date                          13988 non-null  object
 4   Net Due Date (System Calculated Date)  20000 non-null  object
 5   Supplier ID                           20000 non-null  str
 6   Invoice Amount                        20000 non-null  int64
 7   Fiscal year                           20000 non-null  str
 8   Overdue                               20000 non-null  int64
 9   Invoice Status                        20000 non-null  str
 10  Spend Category                        20000 non-null  str
 11  Total Outstanding amount              20000 non-null  int64
 12  Late payment fees                     20000 non-null  int64
 13  Payterm_n                             20000 non-null  int64
 14  Vendor_Type                           20000 non-null  str
dtypes: int64(5), object(4), str(6)
memory usage: 2.3+ MB
```

Просматриваем данные

In [148]:

df_payables_data

Out[148]:

	Invoice Number	Posting Date	Invoice Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
0	INV-5	201	201	30-1	202	S-19	714	201	0	Paid	Taxes	0	0	30	Dom

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	97867560206719 8	9-04-120 00:00:00	9-04-120 00:00:00	2-2019	0-04-010 00:00:00	8	7	9-2020							estic
1	INV-4338	16-03-2	2020-1	NaN	16-04-2	S-1887	85755	2020-2	75	Unpai	RawMat	85755	600	30	Domes

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	5639898658799	020	2-030 00:00:00		020			021		d	erial				tic
2	INV-458688	26-12-2001	24-12-2001	2020-01-0	26-02-2022	S-300	6790	2019-2022	0	Paid	Raw Mater	0	0	60	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	1291138272 1	9	9	100:00:00	0			0			ial				
3	INV-8091675 4	14-01-2020	2020-10-0100:	NaN	14-02-2020	S-497	6575	2020-2021	137	Unpaid	Services	6575	822	30	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	7391140 7		00:00												
4	INV-962372128 73	2019-08-12 00:00:0	2019-04-12 00:00:0	20-01-2020	2020-08-03 00:00:0	S-3110	12635	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	627281	0	0		0										
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	INV-171371909 1	2019-05-12 00:00:	2019-05-12 00:00:	NaN	2020-05-02 00:00:	S-2066	14372	2019-2020	146	Unpaid	Raw Material	14372	876	60	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	5523110	00	00		00										
19966	INV-923263686839	2019-03-06 00:00:00	2019-03-06 00:00:00	20-08-2019	2019-03-09 00:00:00	S-449	13346	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	893	0	0		0										
19977	INV-545534138205547	2020-09-02 00:00:00	2020-04-02 00:00:00	NaN	2020-09-30 00:00:00	S-2322	7437	2020-2021	113	Unpaid	Services	7437	904	30	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
19998	5 INV-5306025960409 2	2019-01-0500:00:00	2019-01-0500:00:00	27-06-2019	2019-01-0800:00:00	S-158	134000	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
19999	INV-35887841858801170	2019-01-03 00:00:00	25-02-2019	13-04-2019	2019-01-06 00:00:00	S-1669	103220	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

20000 rows × 15 columns

Вывод: В датафрейме содержатся данные о кредиторской задолженности клиентов.

Подсчитываем количество пустых значений в столбцах датафрейма

In [149]:

```
df_payables_data.isna().sum()
```

Out[149]:

```
Invoice Number                          0
Posting Date                            0
Invoice Date                            0
Payment Date                         6012
Net Due Date (System Calculated Date)      0
Supplier ID                             0
Invoice Amount                          0
Fiscal year                             0
Overdue                                 0
Invoice Status                          0
Spend Category                          0
Total Outstanding amount                0
Late payment fees                       0
Payterm_n                               0
Vendor_Type                             0
dtype: int64
```

Вывод: Как видно из предыдущего блока, датафрейм df_payables_data содержит 6012 незаполненных данных (NaN) в столбце Payment Date, что свидетельствует о том, что в

данном столбце отсутствует дата последнего платежа клиентом.

Подсчитываем дубликаты

In [150]:

```
df_payables_data.duplicated().sum()
```

Out[150]:

```
np.int64(0)
```

Дублирование строк отсутствует

Как видно из анализа некоторые столбцы с датами содержат не верный формат.
испрвим это. Преобразуем данные из формата object в datetime64

In [151]:

```
df_payables_data['Posting Date'] = pandas.to_datetime(df_payables_data['Posting Date'])
```

In [152]:

```
df_payables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Invoice Number                        20000 non-null  str
 1   Posting Date                          20000 non-null  datetime64[us]
 2   Invoice Date                          20000 non-null  object
 3   Payment Date                          13988 non-null  object
 4   Net Due Date (System Calculated Date)  20000 non-null  object
 5   Supplier ID                           20000 non-null  str
 6   Invoice Amount                        20000 non-null  int64
 7   Fiscal year                           20000 non-null  str
 8   Overdue                               20000 non-null  int64
 9   Invoice Status                        20000 non-null  str
 10  Spend Category                        20000 non-null  str
 11  Total Outstanding amount              20000 non-null  int64
 12  Late payment fees                     20000 non-null  int64
 13  Payterm_n                             20000 non-null  int64
 14  Vendor_Type                           20000 non-null  str
dtypes: datetime64[us](1), int64(5), object(3), str(6)
memory usage: 2.3+ MB
```

In [153]:

```
df_payables_data['Invoice Date'] = pandas.to_datetime(df_payables_data['Invoice Date'])
df_payables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Invoice Number                        20000 non-null  str
 1   Posting Date                          20000 non-null  datetime64[us]
 2   Invoice Date                          20000 non-null  datetime64[us]
 3   Payment Date                          13988 non-null  object
 4   Net Due Date (System Calculated Date)  20000 non-null  object
 5   Supplier ID                           20000 non-null  str
 6   Invoice Amount                        20000 non-null  int64
 7   Fiscal year                           20000 non-null  str
 8   Overdue                               20000 non-null  int64
 9   Invoice Status                        20000 non-null  str
 10  Spend Category                        20000 non-null  str
 11  Total Outstanding amount              20000 non-null  int64
 12  Late payment fees                     20000 non-null  int64
 13  Payterm_n                             20000 non-null  int64
 14  Vendor_Type                           20000 non-null  str
dtypes: datetime64[us](2), int64(5), object(2), str(6)
memory usage: 2.3+ MB
```

In [154]:

```
df_payables_data['Payment Date'] = pandas.to_datetime(df_payables_data['Payment Date'])
df_payables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Invoice Number                        20000 non-null  str
 1   Posting Date                          20000 non-null  datetime64[us]
 2   Invoice Date                          20000 non-null  datetime64[us]
 3   Payment Date                          13988 non-null  datetime64[us]
 4   Net Due Date (System Calculated Date)  20000 non-null  object
 5   Supplier ID                           20000 non-null  str
```

```
 6   Invoice Amount                      20000 non-null  int64
 7   Fiscal year                         20000 non-null  str
 8   Overdue                             20000 non-null  int64
 9   Invoice Status                      20000 non-null  str
 10  Spend Category                      20000 non-null  str
 11  Total Outstanding amount            20000 non-null  int64
 12  Late payment fees                   20000 non-null  int64
 13  Payterm_n                           20000 non-null  int64
 14  Vendor_Type                         20000 non-null  str
dtypes: datetime64[us](3), int64(5), object(1), str(6)
memory usage: 2.3+ MB
```

```
/var/folders/lq/kd04l4wx7tnd4251g2pdj9_40000gn/T/ipykernel_8311/1249894834.py:1:
UserWarning: Parsing dates in %d-%m-%Y format when dayfirst=False (the default) was
specified. Pass `dayfirst=True` or specify a format to silence this warning.
  df_payables_data['Payment Date'] = pandas.to_datetime(df_payables_data['Payment Date'])
```

In [155]:

```
df_payables_data['Net Due Date (System Calculated Date)'] =
pandas.to_datetime(df_payables_data['Net Due Date (System Calculated Date)'])
df_payables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Invoice Number                      20000 non-null  str
 1   Posting Date                        20000 non-null  datetime64[us]
 2   Invoice Date                        20000 non-null  datetime64[us]
 3   Payment Date                        13988 non-null  datetime64[us]
 4   Net Due Date (System Calculated Date)  20000 non-null  datetime64[us]
 5   Supplier ID                         20000 non-null  str
 6   Invoice Amount                      20000 non-null  int64
 7   Fiscal year                         20000 non-null  str
 8   Overdue                             20000 non-null  int64
```

```
 9   Invoice Status                  20000 non-null  str
 10  Spend Category                  20000 non-null  str
 11  Total Outstanding amount        20000 non-null  int64
 12  Late payment fees               20000 non-null  int64
 13  Payterm_n                       20000 non-null  int64
 14  Vendor_Type                     20000 non-null  str
dtypes: datetime64[us](4), int64(5), str(6)
memory usage: 2.3 MB
```

Проверяем данные

In [156]:

```
df_payables_data
```

Out[156]:

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
0	INV-59786756020067198	2019-04-12	2019-04-12	2019-12-30	2020-04-01	S-198	7147	2019-2020	0	Paid	Taxes	0	0	30	Domestic
1	INV-	20	20	Na	20	S-1	85	20	7	Un	Ra	85	60	3	Do

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	4385639898658799	20-03-16	20-12-03	T	20-04-16	87	75	20-2021	5	paid	wMaterial	75	0	0	mestic
2	INV-45	2019-	2019-	2020-	2020-	S-30	679	2019-	0	Paid	RawM	0	0	60	Domes

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	8681291138272121	12-26	12-24	01-01	02-26	0	0	2020			aterial				tic
3	INV-809166	2020-01-1	2020-10-0	NaT	2020-02-1	S-497	6575	2020-2022	137	Unpaid	Services	6575	822	30	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	7547391140 7	4	1		4			1							
4	INV-96237212	2019-08-12	2019-04-12	2020-01-20	2020-08-03	S-3106	126355	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
	873627228														
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19955	INV-171371 19	2019-05-12	2019-05-12	NaT	2020-05-02	S-2006	14372	2019-2020	146	Unpaid	Raw Material	14372	876	60	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
1996	0915523100	INV-923263686	2019-03-06	2019-03-06	2019-08-20	2019-03-09	S-449	1346	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
1997	83989 3	INV-5453413382 0	2020-09-02	2020-04-02	NaT	2020-09-03	S-232	7437	2020-2021	113	Unpaid	Services	7437	904	30	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
1998	5475														
	INV-530606025960400	2019-01-05	2019-01-05	2019-06-27	2019-01-08	S-158	134000	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

	Invoice Number	Posting Date	Invoiced Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
1999	92														
	INV-3588784185880017	2019-01-03	2019-02-25	2019-04-13	2019-01-06	S-1699	103200	2019-2020	0	Paid	Raw Material	0	0	90	Domestic

	Invoice Number	Posting Date	Invoice Date	Payment Date	Net Due Date (System Calculated Date)	Supplier ID	Invoice Amount	Fiscal year	Overdue	Invoice Status	Spend Category	Total Outstanding amount	Late payment fees	Payterm_n	Vendor_Type
0															

20000 rows × 15 columns

In [157]:

```
df_payables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
```

```
 0   Invoice Number                      20000 non-null  str
 1   Posting Date                        20000 non-null  datetime64[us]
 2   Invoice Date                        20000 non-null  datetime64[us]
 3   Payment Date                        13988 non-null  datetime64[us]
 4   Net Due Date (System Calculated Date)  20000 non-null  datetime64[us]
 5   Supplier ID                         20000 non-null  str
 6   Invoice Amount                      20000 non-null  int64
 7   Fiscal year                         20000 non-null  str
 8   Overdue                             20000 non-null  int64
 9   Invoice Status                      20000 non-null  str
 10  Spend Category                      20000 non-null  str
 11  Total Outstanding amount            20000 non-null  int64
 12  Late payment fees                   20000 non-null  int64
 13  Payterm_n                           20000 non-null  int64
 14  Vendor_Type                         20000 non-null  str
dtypes: datetime64[us](4), int64(5), str(6)
memory usage: 2.3 MB
```

Исследуем выбросы

In [158]:

```
df_payables_data.describe()
```

Out[158]:

	Posting Date	Invoice Date	Payment Date	Net Due Date (System Calculated Date)	Invoice Amount	Overdue	Total Outstanding amount	Late payment fees	Payterm_n
count	20000	20000	13988	20000	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000
mean	2019-09-12 17:50:29.760000	2019-09-09 14:59:55.680000	2019-08-29 08:18:58.575922	2019-11-16 04:20:16.800000	11287.620500	33.431150	3389.968650	200.154350	61.005000
min	2019-01-01 00:00:00	2019-01-01 00:00:00	2019-01-02 00:00:00	2019-01-02 00:00:00	101.000000	0.000000	0.000000	0.000000	0.000000
25%	2019-05-10 00:00:00	2019-05-04 00:00:00	2019-05-14 00:00:00	2019-07-02 00:00:00	7295.000000	0.000000	0.000000	0.000000	30.000000
50%	2019-09-09 12:00:00	2019-09-06 00:00:00	2019-08-22 00:00:00	2019-11-16 00:00:00	11237.000000	0.000000	0.000000	0.000000	60.000000
75%	2019-12-16 00:00:00	2019-12-14 00:00:00	2019-11-29 00:00:00	2020-03-07 00:00:00	15624.000000	48.000000	6044.500000	246.000000	90.000000

	Posting Date	Invoice Date	Payment Date	Net Due Date (System Calculated Date)	Invoice Amount	Overdue	Total Outstanding amount	Late payment fees	Payterm_n
max	2020-12-06 00:00:00	2020-12-06 00:00:00	2020-12-06 00:00:00	2020-12-09 00:00:00	19999.000000	266.00000	19981.000000	2560.000000	90.000000
std	NaN	NaN	NaN	NaN	5074.410416	53.949846	5866.871645	362.788222	24.689683

Вывод: Аномалии не выявлены

Анализируем файл receivables_data.xls.

In [159]:

```
df_receivables_data = pandas.read_excel("a/xls/receivables_data.xls")
df_receivables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Business Code          50000 non-null  str
 1   Customer Number        50000 non-null  object
 2   Customer Name          50000 non-null  str
 3   Payment_Date           50000 non-null  object
 4   Business Year          50000 non-null  int64
 5   Posting_Date           50000 non-null  datetime64[us]
 6   Due_Date               50000 non-null  datetime64[us]
 7   Payterm                50000 non-null  int64
 8   Invoice Currency       50000 non-null  str
 9   Total Open Amount      50000 non-null  int64
 10  USD_CURRENNCY          50000 non-null  str
 11  Total Open Amount_USD  50000 non-null  float64
 12  Customer Payment Terms  50000 non-null  str
 13  Invoice ID             49994 non-null  float64
 14  Is Open                50000 non-null  int64
 15  DUNNLEVEL              50000 non-null  int64
 16  Credit_limit           50000 non-null  int64
 17  Baseline_Date          50000 non-null  datetime64[us]
 18  Region                 50000 non-null  str
dtypes: datetime64[us](3), float64(2), int64(6), object(2), str(6)
memory usage: 7.2+ MB
```

In [160]:

```
df_receivables_data
```

Out[160]:

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
0	U001	200769623	WAL-MARcorp	2020-02-11T00:00:00	2020	2020-01-26	2020-02-10	15	USD	54273	USD	54273.280	NAH4	1.930438e+09	0	0	500000	2020-01-23	WEST
1	U001	200980	BENE	2019-	2019	2019-	2019-	20	USD	7965	USD	79656	NAD1	1.929	0	2	5000	2019-	MIDWE

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
		0828		08-08000:00:00		07-22	08-11			7		.600		646e+09			0	07-20	ST
2	U001	200792734	MDV/trust	2019-12-30	2019	2019-09-1	2019-09-2	15	USD	2254	USD	2253.860	NAA8	1.929874e+	0	3	1000000	2019-09-1	NORTHEAST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
3	CA02	1401056866	SYSCllc	01/00/1900 00:00:00	2020	2020-03-30	2020-04-10	11	CAD	33000	USD	24417.78	CA10	2.960623e+09	1	2	1000000	2020-03-26	SOUTHWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
4	U001	200769623	WAL-MARfoundation	2019-11-25 00:00:00	2019	2019-11-13	2019-11-28	15	USD	33133	USD	33133.290	NAH4	1.930148e+09	0	3	100000	2019-11-10	WEST
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
49995	U001	200561861	Coorporation	01/00/1900	2020	2020-04-21	2020-05-06	15	USD	3188	USD	3187.860	NAA8	1.930797e+09	1	3	1000000	2020-04-21	NORTHEAST
49996	U001	200769623	WAL-MARco	2019-09-030	2019	2019-08-15	2019-08-30	15	USD	6767	USD	6766.540	NAH4	1.929744e+0	0	2	1000000	2019-08-11	WEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
49997	U001	0200725595	SAFEWassociates	2020-03-05 00:00:00	2020	2020-02-19	2020-03-05	15	USD	6121	USD	6120.860	NAA8	1.930537e+09	0	1	1000000	2020-02-16	SOUTHEAST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	CreditLimit	Baseline_Date	Region
49998	U001	0072697 9	BJ'S llc	2019-12-12 00:00:00	2019	2019-11-27	2019-12-12	15	USD	63	USD	63.480	NAA8	1.930199e+09	0	0	100000	2019-11-27	MIDWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
49999	U001	0200204311	DECcorp	2019-01-15 00:00:00	2019	2019-01-05	2019-01-24	19	USD	1790	USD	1790.300	NAM4	1.928576e+09	0	1	200000	2019-01-02	SOUTHEAST

50000 rows × 19 columns

Вывод: В файле представлены данные о дебиторской задолженности

Подсчитываем пустые значения в столбцах

In [161]:

```
df_receivables_data.isna().sum()
```

Out[161]:

```
Business Code              0
Customer Number            0
Customer Name              0
Payment_Date               0
Business Year              0
Posting_Date               0
Due_Date                   0
Payterm                    0
Invoice Currency           0
Total Open Amount          0
USD_CURRENNCY              0
Total Open Amount_USD      0
Customer Payment Terms     0
Invoice ID                 6
Is Open                    0
DUNNLEVEL                  0
Credit_limit               0
Baseline_Date              0
Region                     0
dtype: int64
```

Подсчитываем дубликаты

In [162]:

```
df_receivables_data.duplicated().sum()
```

Out[162]:

```
np.int64(36)
```

Есть дубликаты. Убираем их

In [163]:

```
df_receivables_data = df_receivables_data.drop_duplicates(keep=False)
```

Проверяем

In [164]:

```
df_receivables_data.duplicated().sum()
```

Out[164]:

---

np.int64(0)

In [165]:

```
df_receivables_data.info()
```

```
<class 'pandas.DataFrame'>
Index: 49928 entries, 0 to 49999
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Business Code           49928 non-null  str
 1   Customer Number         49928 non-null  object
 2   Customer Name           49928 non-null  str
 3   Payment_Date            49928 non-null  object
 4   Business Year           49928 non-null  int64
 5   Posting_Date            49928 non-null  datetime64[us]
 6   Due_Date                49928 non-null  datetime64[us]
 7   Payterm                 49928 non-null  int64
 8   Invoice Currency        49928 non-null  str
 9   Total Open Amount       49928 non-null  int64
 10  USD_CURRENNCY           49928 non-null  str
 11  Total Open Amount_USD   49928 non-null  float64
 12  Customer Payment Terms  49928 non-null  str
 13  Invoice ID              49922 non-null  float64
 14  Is Open                 49928 non-null  int64
 15  DUNNLEVEL               49928 non-null  int64
 16  Credit_limit            49928 non-null  int64
 17  Baseline_Date           49928 non-null  datetime64[us]
 18  Region                  49928 non-null  str
dtypes: datetime64[us](3), float64(2), int64(6), object(2), str(6)
memory usage: 7.6+ MB
```

In [166]:

```
df_receivables_data
```

Out[166]:

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	PostingDate	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
0	U001	200769623	WAL-MARcorp	2020-02-11 00:	2020	2020-01-26	2020-02-10	15	USD	54273	USD	54273.280	NAH4	1.930438e+09	0	0	50000	2020-01-23	WEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
1	U001	200980828	BENE	2019-08-08 00:00:00	2019	2019-07-22	2019-08-11	20	USD	79657	USD	79656.600	NAD1	1.92964e+09	0	2	50000	2019-07-20	MIDWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
2	U001	0200792734	MDV/trust	2019-12-30 00:00:00	2019	2019-09-14	2019-09-29	15	USD	2254	USD	2253.860	NAA8	1.929874e+09	0	3	100000	2019-09-14	NORTHEAST
3	CA	140	SY	01	20	20	20	1	CA	33	US	24	CA	2.	1	2	10	20	SO

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
	02	0105686	SCllc	/00/1900	20	20-03-30	20-04-10	1	D	00	D	41.778	10	9.606236e+09			0000	20-03-26	UTHWEST
4	U001	2007696223	WAL-MARfounda	2019-11-25 00:	2019	2019-11-13	2019-11-28	15	USD	331333	USD	331333.290	NAH4	1.930148e+09	0	3	1000000	2019-11-10	WEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
49995	U001	0200561861	COcorporation	01/00/1900 00:00	2020	2020-04-21	2020-05-06	15	USD	3188	USD	3187.860	NAA8	1.930797e+09	1	3	100000	2020-04-21	NORTHEAST
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
4996	U001	0200769623	WAL-MARco	2019-09-03000:00	2019	2019-08-15	2019-08-30	15	USD	6767	USD	6766.540	NAH4	1.929744e+09	0	2	100000	2019-08-11	WEST
4999	U001	0200777	SAFEW	2020-	2020	2020-	2020-	15	USD	61211	USD	61220.	NAA8	1.930	0	1	10000	2020-	SOUTH

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
7		2595	associates	03-05000:00:00		02-19	03-05					860		537e+09			0	02-16	EAST
49998	U001	0072697 9	BJ'S llc	2019-12-12	2019	2019-11-2	2019-12-1	15	USD	63	USD	63.480	NAA8	1.93019 9e+	0	0	1000000	2019-11-2	MIDWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
49999	U001	0200204031	DECcorp	2019-01-15 00:00:00	2019	2019-01-05	2019-01-24	19	USD	1790	USD	1790.300	NAM4	1.928576e+09	0	1	200000	2019-01-02	SOUTHEAST
				00:00:00		7	2							09				7	

BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

49928 rows × 19 columns

Преобразование столбцов с датами

При стандарном выполнеии запроса выходит ошибка

```
df_receivables_data['Payment_Date'] =
pandas.to_datetime(df_receivables_data['Payment_Date'])
df_receivables_data.info()
```

DateParseError Traceback (most recent call last) Cell In[57], line 1 ----> 1 df_receivables_data['Payment_Date'] = pandas.to_datetime(df_receivables_data['Payment_Date']) 2 df_receivables_data.info()

File ~/Documents/КПК/Чемпионат/2026/машинное обучение/Юниоры/work/env/lib/python3.12/site-packages/pandas/core/tools/datetimes.py:1036, in to_datetime(arg, errors, dayfirst, yearfirst, utc, format, exact, unit, origin, cache) 1034 result = arg.tz_localize("utc") 1035 elif isinstance(arg, ABCSeries): -> 1036 cache_array = _maybe_cache(arg, format, cache, convert_listlike) 1037 if not cache_array.empty: 1038 result = arg.map(cache_array)

File ~/Documents/КПК/Чемпионат/2026/машинное обучение/Юниоры/work/env/lib/python3.12/site-packages/pandas/core/tools/datetimes.py:254, in _maybe_cache(arg, format, cache, convert_listlike) 252 unique_dates = unique(arg) 253 if len(unique_dates) < len(arg): --> 254 cache_dates = convert_listlike(unique_dates, format) 255 # GH#45319 256 try:

File ~/Documents/КПК/Чемпионат/2026/машинное обучение/Юниоры/work/env/lib/python3.12/site-packages/pandas/core/tools/datetimes.py:437, in _convert_listlike_datetimes(arg, format, name, utc, unit, errors, dayfirst, yearfirst, exact) 434 if format is not None and format != "mixed": 435 return _array_strptime_with_fallback(arg, name, utc, format, exact, errors) --> 437 result, tz_parsed = objects_to_datetime64(438 arg, ... File pandas/_libs/tslibs/parsing.pyx:307, in pandas._libs.tslibs.parsing.parse_datetime_string()

File pandas/_libs/tslibs/parsing.pyx:218, in pandas._libs.tslibs.parsing._parse_delimited_date()

DateParseError: Invalid date specified (1/0) Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Данная ошибка говорит о том, что в форматированном столбце есть несоответствие формату даты. Нужно исправить.

In [167]:

df_receivables_data['Payment_Date'] **=**

```
pandas.to_datetime(df_receivables_data['Payment_Date'], errors='coerce',
format="%y%m%d")
```

Проверяем

In [168]:

```
df_receivables_data.info()
```

```
<class 'pandas.DataFrame'>
Index: 49928 entries, 0 to 49999
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Business Code           49928 non-null  str
 1   Customer Number         49928 non-null  object
 2   Customer Name           49928 non-null  str
 3   Payment_Date            39942 non-null  datetime64[us]
 4   Business Year           49928 non-null  int64
 5   Posting_Date            49928 non-null  datetime64[us]
 6   Due_Date                49928 non-null  datetime64[us]
 7   Payterm                 49928 non-null  int64
 8   Invoice Currency        49928 non-null  str
 9   Total Open Amount       49928 non-null  int64
 10  USD_CURRENNCY           49928 non-null  str
 11  Total Open Amount_USD   49928 non-null  float64
 12  Customer Payment Terms  49928 non-null  str
 13  Invoice ID              49922 non-null  float64
 14  Is Open                 49928 non-null  int64
 15  DUNNLEVEL               49928 non-null  int64
 16  Credit_limit            49928 non-null  int64
 17  Baseline_Date           49928 non-null  datetime64[us]
 18  Region                  49928 non-null  str
dtypes: datetime64[us](4), float64(2), int64(6), object(1), str(6)
```

memory usage: 7.6+ MB

In [169]:

df_receivables_data

Out[169]:

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
0	U001	200769622	WAL-MARco	2020-02-1	2020	2020-01-2	2020-02-1	15	USD	54273	USD	54273.28	NAH4	1.930438e	0	0	500000	2020-01-2	WEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
		3	rp	1		6	0					0		+09				3	
1	U001	2009808828	BENE	2019-08-08	2019	2019-07-22	2019-08-11	20	USD	796557	USD	79656.600	NAD1	1.929646e+09	0	2	500000	2019-07-20	MIDWEST
2	U001	2007792	MDV/tr	2019-1	2019	2019-0	2019-0	15	USD	2254	USD	2253.8	NAA8	1.92298	0	3	100000	2019-0	NORTHE

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
		734	ust	2-30		9-14	9-29					60		74e+09			0	9-14	AST
3	CA02	140105686	SYSCllc	NaT	2020	2020-03-30	2020-04-10	11	CAD	3300	USD	2441.778	CA10	2.960623e+09	1	2	100000	2020-03-26	SOUTHWEST
4	U000	200	WAL	2011	2011	2011	2011	15	US	3311	US	331	NAH	1.9	0	3	100	2011	WES

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
	1	769623	-MARfoundation	9-11-25	9	9-11-13	9-11-28		D	33	D	33.290	4	3.0148e+09			000	9-11-10	T
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4999	U001	200056	Cocor	NaT	2020	2020--	2020-	15	USD	31888	USD	31887.	NAA8	1.930	1	3	100000	2020-	NORTH

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
5		1861	poration			04-21	05-06					860		797e+09			0	04-21	EAST
49996	U001	200769623	WAL-MARco	2019-09-03	2019	2019-08-15	2019-08-30	15	USD	67667	USD	6766.540	NAH4	1.929744e+09	0	2	100000	2019-08-11	WEST
49	U0	20	SA	20	20	20	20	1	US	61	US	61	NA	1.	0	1	10	20	SO

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
997	01	0772595	FEWassociates	20-03-05	20	20-02-19	20-03-05	5	D	21	D	20.860	A8	9.305376e+09			0000	20-02-16	UTHEAST
49998	U001	200726979	BJ's llc	2019-12-1	2019	2019-11-2	2019-12-1	15	USD	63	USD	63.480	NAA8	1.930199e+	0	0	1000000	2019-11-2	MIDWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
49999	U001	0000204311	DECcorp	2019-01-15	2019	2019-01-05	2019-01-24	19	USD	1790	USD	1790.300	NAM4	1.928576e+09	0	1	200000	2019-01-02	SOUTHEAST

49928 rows × 19 columns

Исследуем выбросы

In [170]:

```
df_receivables_data.describe()
```

Out[170]:

	Payment_Date	Business Year	Posting_Date	Due_Date	Payterm	Total Open Amount	Total Open Amount_USD	Invoice ID	Is Open	DUNNLEVEL	Credit_limit	Baseline_Date
count	39942	49928.000000	4928	4928	49928.000000	49928.000000	49928.000000	4.992200e+04	49928.000000	49928.000000	49928.000000	4928
mean	2019-08-10 08:25:46.642631	2019.305660	2019-09-11 20:44:15.151418	2019-09-28 23:26:27.437910	17.112642	32331.848161	31186.625573	2.011251e+09	0.200008	1.503725	78110.679378	2019-09-09 20:54:45.050472
mi	2019-	2019.	2018-	2018-	-50.00	1.000	0.720	1.928	0.000	0.000	0.000	2018-

	Payment_Date	Business Year	Posting_Date	Due_Date	Payterm	Total Open Amount	Total Open Amount_USD	Invoice ID	Is Open	DUNNLEVEL	Credit_limit	Baseline_Date
n	01-03 00:00:00	0000	12-30 00:00:00	12-24 00:00:00	0000	000	00	502e+09	000	000	000	12-26 00:00:00
25%	2019-05-01 00:00:00	2019.0000	2019-05-06 00:00:00	2019-05-24 00:00:00	15.000000	4924.000000	4776.510700	1.92934 2e+09	0.0000000	1.0000000	5000.0000000	2019-05-04 00:00:00
50%	2019-08-07 00:00:00	2019.0000	2019-09-09 00:00:00	2019-09-26 00:00:00	15.000000	17604.000000	17296.311550	1.92996 4e+09	0.0000000	2.0000000	10000.000000	2019-09-07 00:00:00
75%	2019-11-15 00:00:00	2020.0000	2020-01-31 00:00:00	2020-02-16 00:00:00	15.000000	4712 4.000000	4613 1.86750	1.93061 9e+09	0.0000000	3.0000000	10000 00.000000	2020-01-29 00:00:00
max	2020-	2020.	2020-	2020-	120.0	6685	6685	2.960	1.000	3.000	5000	2020-

	Payment_Date	Business Year	Posting_Date	Due_Date	Payterm	Total Open Amount	Total Open Amount_USD	Invoice ID	Is Open	DUNNLEVEL	Credit_limit	Baseline_Date
	05-22 00:00:00	0000000	05-22 00:00:00	07-10 00:00:00	00000	93.000000	93.360000	636e+09	000	000	00.000000	05-19 00:00:00
std	NaN	0.460691	NaN	NaN	10.269722	392.656296	361.071550	2.764918e+08	0.400010	1.118057	4298.133921	NaN

Вывод: Столбец 'Payterm' (Срок выплат) содержит минимальное отрицательное значение, что не может быть. Высним сколько всего отрицательных значений в этом столбце

In [171]:

```
df_receivables_data[df_receivables_data['Payterm'] < 0]
```

Out[171]:

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
183	U001	200416837	DECus	2019-03-25	2019	2019-03-25	2019-03-23	-2	USD	20857	USD	20857.19	NAM1	1.929024e+09	0	3	200000	2019-03-21	SOUTHWEST
412	U001	200706844	WINCco	2019-02-19	2019	2019-02-01	2018-12-29	-34	USD	577	USD	576.88	NAA8	1.928721e+0	0	3	100000	2019-01-29	SOUTHEAST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
627	U001	0200424504	Min	NaT	2020	2020-04-28	2020-04-15	-13	USD	2428	USD	2428.04	NACG	1.930830e+09	1	0	50000	2020-04-28	SOUTHWEST
816	U001	0200455332	PEAco	2019-03-	2019	2019-02-	2019-02-	-5	USD	11798	USD	11797.9	NAD5	1.928793	0	0	50000	2019-02-	MIDWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
		2		11		13	08					8		e+09				11	
1774	U001	200592182	DECAcorporation	NaT	2020	2020-03-24	2020-03-23	-1	USD	361103	USD	361023.76	NAM1	1.930692e+09	1	3	200000	2020-03-24	SOUTHWEST
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
48093	U001	0002004031	DECco	2019-05-25	2019	2019-05-25	2019-05-23	-2	USD	12776	USD	12776.45	NAM1	1.92937e+09	0	0	100000	2019-05-24	SOUTHEAST
48360	U001	0004168683	DECcorp	2019-01-	2019	2019-01-	2019-01-	-2	USD	9774	USD	9773.84	NAM2	1.92869e+09	0	0	200000	2019-01-	SOUTHWES

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
		7		28		28	26							e+09				26	T
49400	U001	2008037200	DECtrust	2020-01-29	2020	2020-01-27	2020-01-26	-1	USD	3191	USD	31904.60	NAM2	1.930444e+09	0	0	200000	2020-01-26	MIDWEST
49580	U001	200880	DECsy	2020-	2020	2020-	2020-	-1	USD	85118	USD	85118.	NAM2	1.930	0	3	5000	2020-	MIDWE

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
9		3720	stems	01-27		01-27	01-26					44		445e+09			0	01-27	ST
49592	U001	200592182	DECAus	2019-08-28	2019	2019-08-28	2019-08-23	-5	USD	187225	USD	187244.67	NAM1	1.929802e+09	0	1	200000	2019-08-25	SOUTHWEST

140 rows × 19 columns

Всего таких аномальных строк 140. Удалим их

In [172]:

```
df_receivables_data = df_receivables_data[df_receivables_data['Payterm'] > 0]
```

In [173]:

```
df_receivables_data
```

Out[173]:

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
0	U001	200769623	WAL-MARcorp	2020-02-11	2020	2020-01-26	2020-02-10	15	USD	54273	USD	54273.280	NAH4	1.930438e+09	0	0	500000	2020-01-23	WEST
1	U001	200980828	BENE	2019-08-08	2019	2019-07-22	2019-08-11	20	USD	79657	USD	79656.600	NAD1	1.929646e+0	0	2	500000	2019-07-20	MIDWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
2	U001	0200792734	MDV/trust	2019-12-30	2019	2019-09-14	2019-09-29	15	USD	2254	USD	2253.860	NAA8	1.929874e+09	0	3	1000000	2019-09-14	NORTHEAST
3	CA02	140105568	SYSCllc	Nat	2020	2020-03-26	2020-04-10	11	CAD	3300	USD	2441.77	CA10	2.960623e+09	1	2	1000000	2020-03-26	SOUTHWEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
		6				30	10					8		e+09				26	T
4	U001	200769623	WAL-MARfoundation	2019-11-25	2019	2019-11-13	2019-11-28	15	USD	331333	USD	331333.290	NAH4	1.930148e+09	0	3	100000	2019-11-10	WEST
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
49995	U001	200561861	COcorporation	NaT	2020	2020-04-21	2020-05-06	15	USD	3188	USD	3187.860	NAA8	1.930797e+09	1	3	100000	2020-04-21	NORTHEAST
49996	U001	200769 6	WAL-MAR	2019-09	2019	2019-08	2019-08	15	USD	6767	USD	6766.54	NAH4	1.929 74	0	2	100000	2019-08	WEST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
		23	co	-03		-15	-30					0		4e+09				-11	
49997	U001	200772595	SAFEWassociates	2020-03-05	2020	2020-02-19	2020-03-05	15	USD	61211	USD	6120.860	NAA8	1.930537e+09	0	1	1000000	2020-02-16	SOUTHEAST

	BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region
49998	U001	200726979	BJ's llc	2019-12-12	2019	2019-11-27	2019-12-12	15	USD	63	USD	63.4800	NAA8	1.930199e+09	0	0	100000	2019-11-27	MIDWEST
49999	U001	200020431	DECcorp	2019-01-15	2019	2019-01-05	2019-01-24	19	USD	1790	USD	1790.300	NAM4	1.928576e+0	0	1	200000	2019-01-02	SOUTHEAST

BusinessCode	CustomerNumber	CustomerName	Payment_Date	BusinessYear	Posting_Date	Due_Date	Payterm	InvoiceCurrency	TotalOpenAmount	USD_CURRENNCY	TotalOpenAmount_USD	CustomerPaymentTerms	InvoiceID	IsOpen	DUNNLEVEL	Credit_limit	Baseline_Date	Region

49077 rows × 19 columns

Вывод: Как видим объем датафрейма уменьшился с 49928 до 49077

Анализируем файл suppliers_data.xls

In [174]:

```
df_suppliers_data = pandas.read_excel("a/xls/suppliers_data.xls")
```

```
df_suppliers_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Supplier ID       500 non-null    str
 1   Supplier Name     500 non-null    str
 2   Payment Terms     500 non-null    str
 3   Vendor Type       500 non-null    str
 4   Supplier Category  500 non-null    str
dtypes: str(5)
memory usage: 19.7 KB
```

In [175]:

```
df_suppliers_data
```

Out[175]:

	Supplier ID	Supplier Name	Payment Terms	Vendor Type	Supplier Category
0	S-281	Roth-Sanchez	Net 60	Domestic	Raw Material
1	S-438	Peterson Inc	Net 60	Domestic	Raw Material

	Supplier ID	Supplier Name	Payment Terms	Vendor Type	Supplier Category
2	S-480	Morton, Newman and Baker	Net 90	Domestic	Services
3	S-148	Evans Inc	Net 30	Domestic	Utility
4	S-8	Hart Ltd	Net 90	International	Taxes
...	...	...	...	...	...
495	S-345	Terrell-Wyatt	Net 60	Domestic	Raw Material
496	S-132	Porter-Anderson	Net 60	Domestic	Raw Material
497	S-451	Brown-Fisher	Net 30	Domestic	Raw Material
498	S-235	Warren and Sons	Net 90	Domestic	Raw Material
499	S-321	Pollard Group	Net 90	Domestic	Raw Material

500 rows × 5 columns

Подсчитываем сумму пустых значений в столбцах датафрейма

In [176]:

```
df_suppliers_data.isna().sum()
```

Out[176]:

```
Supplier ID          0
Supplier Name        0
Payment Terms        0
Vendor Type          0
Supplier Category    0
dtype: int64
```

Считаем дубликаты

In [177]:

```
df_suppliers_data.duplicated().sum()
```

Out[177]:

```
np.int64(0)
```

Выбросы

In [178]:

```
df_customers_data.describe()
```

Out[178]:

	Credit Limit
count	4574.000000
mean	89435.942282
std	178507.610874
min	5000.000000
25%	20000.000000
50%	50000.000000
75%	100000.000000
max	1000000.000000

Вывод: Аномалии не обнаружены

Загрузка данных в базу данных

Пример для postgreSQL

```
connection =
psycopg2.connect(database="tourism",user="postgres",password="admin",port
=5432,host="localhost")
cur = connection.cursor()
```

Импорт необходимых библиотек

In [179]:

```
import sqlite3
import pandas
```

Используем четыре обработанных датафрейма для загрузки в четыре разных таблицы (база данных одна)

Создаем таблицу customers на основе датафрейма df_customers_data

In [180]:

```
df_customers_data.info()
```

```
<class 'pandas.DataFrame'>
Index: 4574 entries, 0 to 4976
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Customer ID           4574 non-null   str
 1   Customer Name         4574 non-null   str
 2   Customer Payment Terms  4574 non-null   str
 3   Address               4574 non-null   str
 4   Credit Limit          4574 non-null   int64
dtypes: int64(1), str(4)
memory usage: 214.4 KB
```

In [181]:

```
connection = sqlite3.connect("database.db")
cur = connection.cursor()

customers_data = df_customers_data .values.tolist()

sql_create_table_customers = """
CREATE TABLE IF NOT EXISTS customers(
        "Customer ID" TEXT PRIMARY KEY ,
        "Customer_Name" TEXT,
        "Customer_Payment_Terms" TEXT,
        "Address" TEXT,
        "Credit_Limit" INTEGER
);
"""
```

```
cur.execute(sql_create_table_customers)
connection.commit()

for i, row in enumerate(customers_data):
        sql_insert_customers = """
                INSERT INTO customers(
                        "Customer ID",
                        "Customer_Name",
                        "Customer_Payment_Terms",
                        "Address",
                        "Credit_Limit"
                        )
                        VALUES (?, ?, ?, ?, ?);
        """
        cur.execute(sql_insert_customers, (row[0], row[1], row[2], row[3], row[4]))
        connection.commit()

connection.close()
```

In [182]:

```
df_payables_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                               --------------  -----
 0   Invoice Number                          20000 non-null  str
 1   Posting Date                           20000 non-null  datetime64[us]
 2   Invoice Date                           20000 non-null  datetime64[us]
 3   Payment Date                            13988 non-null  datetime64[us]
 4   Net Due Date (System Calculated Date)  20000 non-null  datetime64[us]
 5   Supplier ID                           20000 non-null  str
 6   Invoice Amount                          20000 non-null  int64
 7   Fiscal year                           20000 non-null  str
 8   Overdue                               20000 non-null  int64
 9   Invoice Status                         20000 non-null  str
 10  Spend Category                          20000 non-null  str
```

```
 11  Total Outstanding amount              20000 non-null  int64
 12  Late payment fees                     20000 non-null  int64
 13  Payterm_n                             20000 non-null  int64
 14  Vendor_Type                           20000 non-null  str
dtypes: datetime64[us](4), int64(5), str(6)
memory usage: 2.3 MB
```

In [183]:

```
connection = sqlite3.connect("database.db")
cur = connection.cursor()

payables_data = df_payables_data .values.tolist()

sql_create_table_payables = """
CREATE TABLE IF NOT EXISTS payables(
        "Invoice Number" TEXT PRIMARY KEY,
        "Posting Date" TIMESTAMP,
        "Invoice Date" TIMESTAMP,
        "Payment Date" TIMESTAMP,
        "Net Due Date (System Calculated Date)" TIMESTAMP,
        "Supplier ID" TEXT,
        "Invoice Amount" INTEGER,
        "Fiscal year" TEXT,
        "Overdue" INTEGER,
        "Invoice Status" TEXT,
        "Spend Category" TEXT,
        "Total Outstanding amount" INTEGER,
        "Late payment fees" INTEGER,
        "Payterm_n" INTEGER,
        "Vendor_Type" TEXT
);
"""

cur.execute(sql_create_table_payables)
connection.commit()

for i, row in enumerate(payables_data):
        sql_insert_payables = """
                INSERT INTO payables(
                        "Invoice Number",
                        "Posting Date",
                        "Invoice Date",
```

```
                    "Payment Date",
                    "Net Due Date (System Calculated Date)",
                    "Supplier ID",
                    "Invoice Amount",
                    "Fiscal year",
                    "Overdue",
                    "Invoice Status",
                    "Spend Category",
                    "Total Outstanding amount",
                    "Late payment fees",
                    "Payterm_n",
                    "Vendor_Type"
                    )
                    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
        """
        cur.execute(sql_insert_payables, (row[0], str(row[1]), str(row[2]), str(row[3]), str(row[4]),
row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], row[14]))
        connection.commit()

connection.close()
```

In [184]:

```
df_receivables_data.info()
```

```
<class 'pandas.DataFrame'>
Index: 49077 entries, 0 to 49999
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Business Code         49077 non-null  str
 1   Customer Number       49077 non-null  object
 2   Customer Name         49077 non-null  str
 3   Payment_Date          39195 non-null  datetime64[us]
 4   Business Year         49077 non-null  int64
 5   Posting_Date          49077 non-null  datetime64[us]
 6   Due_Date              49077 non-null  datetime64[us]
 7   Payterm               49077 non-null  int64
 8   Invoice Currency      49077 non-null  str
```

```
 9   Total Open Amount       49077 non-null  int64
 10  USD_CURRENNCY           49077 non-null  str
 11  Total Open Amount_USD   49077 non-null  float64
 12  Customer Payment Terms  49077 non-null  str
 13  Invoice ID              49077 non-null  float64
 14  Is Open                 49077 non-null  int64
 15  DUNNLEVEL               49077 non-null  int64
 16  Credit_limit            49077 non-null  int64
 17  Baseline_Date           49077 non-null  datetime64[us]
 18  Region                  49077 non-null  str
dtypes: datetime64[us](4), float64(2), int64(6), object(1), str(6)
memory usage: 7.5+ MB
```

In [185]:

```
connection = sqlite3.connect("database.db")
cur = connection.cursor()

receivables_data = df_receivables_data .values.tolist()

sql_create_table_receivables = """
CREATE TABLE IF NOT EXISTS receivables(
        "Business Code" TEXT,
        "Customer Number" TEXT,
        "Customer Name" TEXT,
        "Payment_Date" TIMESTAMP,
        "Business Year" INTEGER,
        "Posting_Date" TIMESTAMP,
        "Due_Date" TIMESTAMP,
        "Payterm" INTEGER,
        "Invoice Currency" TEXT,
        "Total Open Amount" INTEGER,
        "USD_CURRENNCY" TEXT,
        "Total Open Amount_USD" FLOAT,
        "Customer Payment Terms" TEXT,
        "Invoice ID" FLOAT,
        "Is Open" INTEGER,
        "DUNNLEVEL" INTEGER,
        "Credit_limit" INTEGER,
        "Baseline_Date" TIMESTAMP,
        "Region" TEXT
);
"""
```

```
cur.execute(sql_create_table_receivables)
connection.commit()

for i, row in enumerate(receivables_data):
        sql_insert_receivables = """
                INSERT INTO receivables(
                        "Business Code",
                        "Customer Number",
                        "Customer Name",
                        "Payment_Date",
                        "Business Year",
                        "Posting_Date",
                        "Due_Date",
                        "Payterm",
                        "Invoice Currency",
                        "Total Open Amount",
                        "USD_CURRENNCY",
                        "Total Open Amount_USD",
                        "Customer Payment Terms",
                        "Invoice ID",
                        "Is Open",
                        "DUNNLEVEL",
                        "Credit_limit",
                        "Baseline_Date",
                        "Region"
                        )
                        VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?);
        """
        cur.execute(sql_insert_receivables, (row[0], row[1], row[2], str(row[3]), row[4],
str(row[5]), str(row[6]), row[7], row[8], row[9], row[10], row[11], row[12], row[13], row[14],
row[15], row[16], str(row[17]), row[18]))
        connection.commit()

connection.close()
```

In [186]:

```
df_suppliers_data.info()
```

```
<class 'pandas.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Supplier ID       500 non-null    str
 1   Supplier Name     500 non-null    str
 2   Payment Terms     500 non-null    str
 3   Vendor Type       500 non-null    str
 4   Supplier Category  500 non-null    str
dtypes: str(5)
memory usage: 19.7 KB
```

In [187]:

```
connection = sqlite3.connect("database.db")
cur = connection.cursor()

suppliers_data = df_suppliers_data .values.tolist()

sql_create_table_suppliers = """
CREATE TABLE IF NOT EXISTS suppliers(
        "Supplier ID" TEXT PRIMARY KEY,
        "Supplier Name" TEXT,
        "Payment Terms" TEXT,
        "Vendor Type" TEXT,
        "Supplier Category" TEXT
);
"""

cur.execute(sql_create_table_suppliers)
connection.commit()

for i, row in enumerate(customers_data):
        sql_insert_suppliers = """
                INSERT INTO suppliers(
                        "Supplier ID",
                        "Supplier Name",
                        "Payment Terms",
                        "Vendor Type",
                        "Supplier Category"
                        )
```

```
                    VALUES (?, ?, ?, ?, ?);
        """
        cur.execute(sql_insert_suppliers, (row[0], row[1], row[2], row[3], row[4]))
        connection.commit()

connection.close()
```