In [117]:
```
import pandas as pd
```

In [118]:
```
import numpy as np
```

In [119]:
```
data1=pd.read_csv("/home/palacement/Downloads/basket_details.csv")
```

In [120]:
```
data=pd.read_csv("/home/palacement/Downloads/customer_details.csv")
```

In [121]:
```
data.describe()
```

Out[121]:

	customer_id	customer_age	tenure
count	2.000000e+04	20000.000000	20000.000000
mean	1.760040e+07	262.222550	44.396800
std	8.679505e+06	604.321589	31.998376
min	2.093000e+03	-34.000000	4.000000
25%	1.188115e+07	29.000000	21.000000
50%	1.560912e+07	38.000000	35.000000
75%	2.228484e+07	123.000000	60.000000
max	4.462566e+07	2022.000000	133.000000

In [122]:
```
data1.describe()
```

Out[122]:

	customer_id	product_id	basket_count
count	1.500000e+04	1.500000e+04	15000.000000
mean	1.808567e+07	3.269771e+07	2.153733
std	1.233000e+07	1.629455e+07	0.517929
min	4.784000e+03	4.939000e+04	2.000000
25%	8.659327e+06	3.137412e+07	2.000000
50%	1.520775e+07	3.694759e+07	2.000000
75%	2.663904e+07	4.502408e+07	2.000000
max	4.460824e+07	5.579097e+07	10.000000

In [123]:
```
data.info() and data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   customer_id   20000 non-null  int64
 1   sex           20000 non-null  object
 2   customer_age  20000 non-null  float64
 3   tenure        20000 non-null  int64
dtypes: float64(1), int64(2), object(1)
memory usage: 625.1+ KB
```

In [124]:
```
data.tail()
```

Out[124]:

	customer_id	sex	customer_age	tenure
19995	12557307	Male	41.0	52
19996	12595961	Male	29.0	52
19997	12520991	Male	35.0	52
19998	12612719	Male	39.0	52
19999	12572063	Male	28.0	52

```
data1.groupby(['customer_id']).count()
```

In [125]:
```
data1.groupby(['customer_id']).count()
```

Out[125]:

customer_id	product_id	basket_date	basket_count
4784	1	1	1
8314	2	2	2
8857	1	1	1
9273	1	1	1
11172	1	1	1
...	...	...	...
44460516	1	1	1
44461180	1	1	1
44473609	1	1	1
44486815	1	1	1
44608245	1	1	1

13871 rows × 3 columns

In [126]:
```
data.groupby(['customer_id']).count()
```

Out[126]:

customer_id	sex	customer_age	tenure
2093	1	1	1
12817	1	1	1
14309	1	1	1
15155	1	1	1
23205	1	1	1
...	...	...	...
44392831	1	1	1
44401175	1	1	1
44431821	1	1	1
44621778	1	1	1
44625658	1	1	1

20000 rows × 3 columns

In [127]:
```
data.groupby(['customer_age']).count()
```

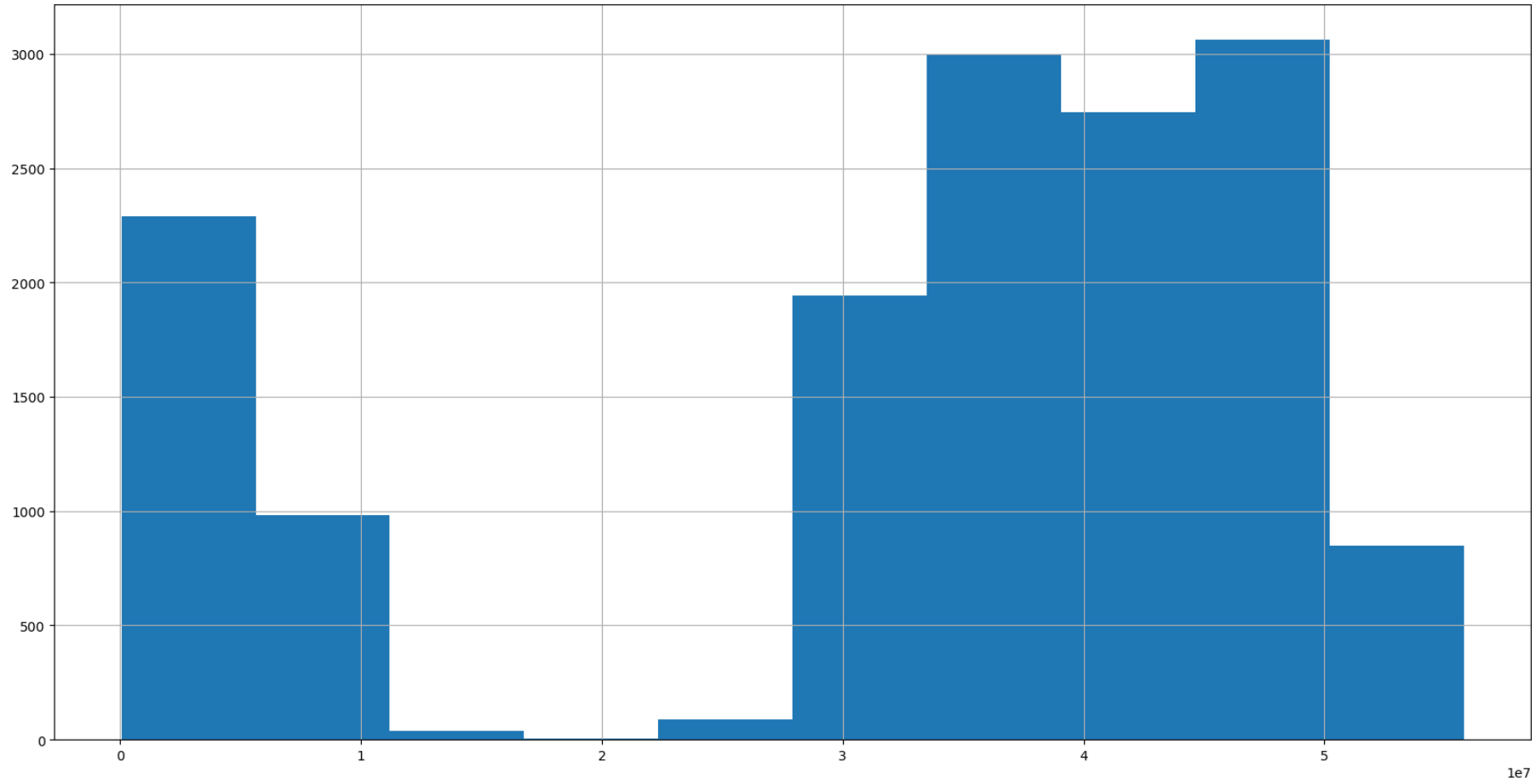
Out[127]:

customer_age	customer_id	sex	tenure
-34.0	1	1	1
3.0	2	2	2
4.0	1	1	1
5.0	710	710	710
6.0	1	1	1
...	...	...	...
127.0	1	1	1
130.0	1	1	1
139.0	1	1	1
149.0	1	1	1
2022.0	2102	2102	2102

93 rows × 3 columns

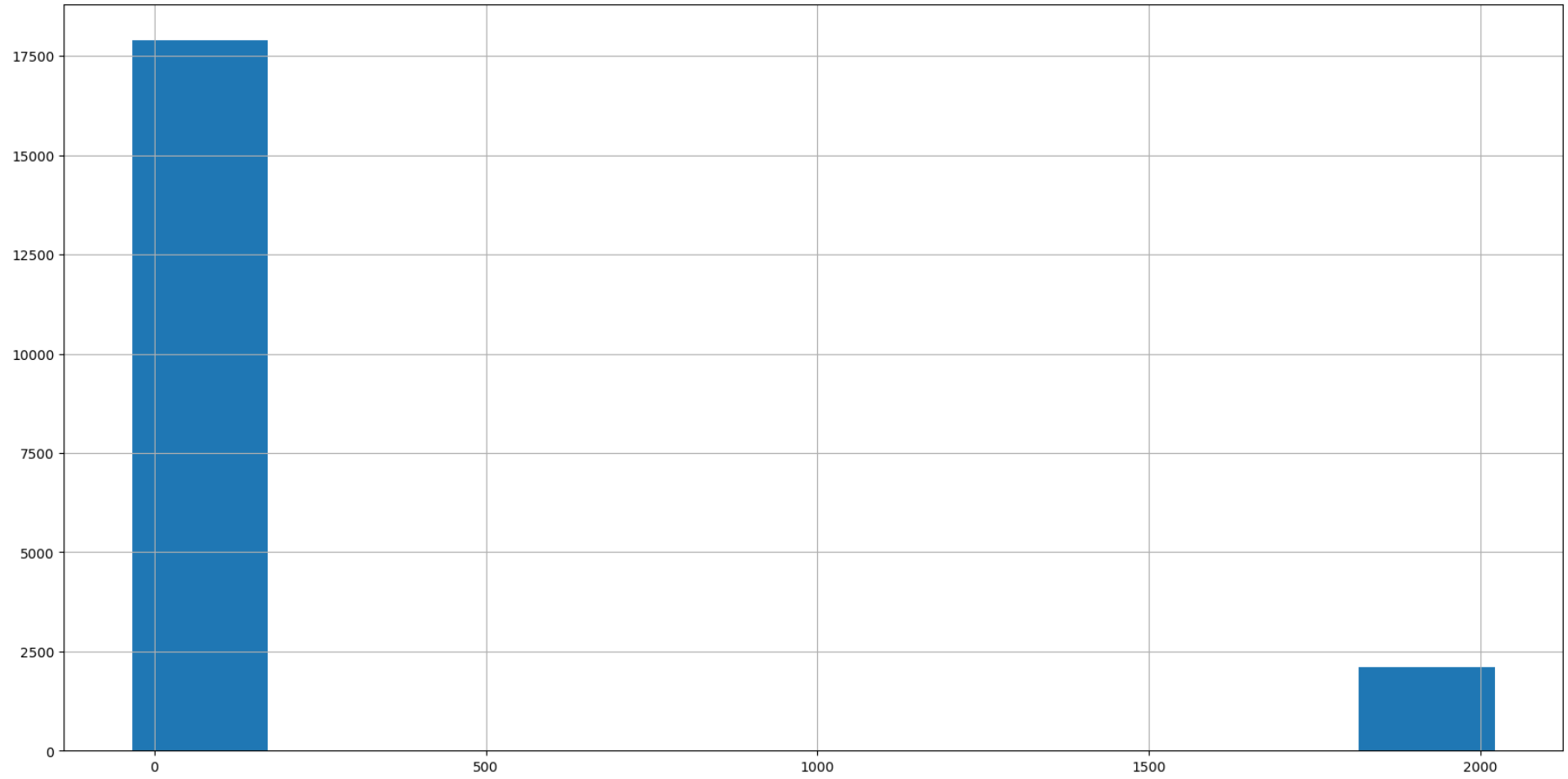
In [128]:
```
data1['product_id'].hist(figsize=(20,10))
```

Out[128]: <Axes: >



In [129]:
```
data['customer_age'].hist(figsize=(20,10))
```

Out[129]: <Axes: >



In [ ]:
```

```

In [130]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in ./anaconda3/lib/python3.10/site-packages (0.12.2)
Requirement already satisfied: pandas>=0.25 in ./anaconda3/lib/python3.10/site-packages (from seaborn) (1.
5.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in ./anaconda3/lib/python3.10/site-packages (from se
aborn) (3.7.0)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in ./anaconda3/lib/python3.10/site-packages (from seabo
rn) (1.23.5)
Requirement already satisfied: pyparsing>=2.3.1 in ./anaconda3/lib/python3.10/site-packages (from matplotli
b!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in ./anaconda3/lib/python3.10/site-packages (from matpl
otlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in ./anaconda3/lib/python3.10/site-packages (from matplotlib!=
3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in ./anaconda3/lib/python3.10/site-packages (from matplotl
ib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in ./anaconda3/lib/python3.10/site-packages (from matplotli
b!=3.6.1,>=3.1->seaborn) (22.0)
Requirement already satisfied: cycler>=0.10 in ./anaconda3/lib/python3.10/site-packages (from matplotlib!=
3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in ./anaconda3/lib/python3.10/site-packages (from matplotli
b!=3.6.1,>=3.1->seaborn) (1.0.5)
Requirement already satisfied: fonttools>=4.22.0 in ./anaconda3/lib/python3.10/site-packages (from matplotl
ib!=3.6.1,>=3.1->seaborn) (4.25.0)
Requirement already satisfied: pytz>=2020.1 in ./anaconda3/lib/python3.10/site-packages (from pandas>=0.25-
>seaborn) (2022.7)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.10/site-packages (from python-dateutil>=
2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [131]:
```
test=pd.merge(data1,data,on ="customer_id")
```

In [132]:
```
test
```

Out[132]:

	customer_id	product_id	basket_date	basket_count	sex	customer_age	tenure
0	4897641	34525548	2019-06-15	2	Male	40.0	114
1	11623549	50394038	2019-06-18	2	Male	30.0	63
2	11665521	41476812	2019-06-15	2	Female	51.0	62
3	4193819	6455162	2019-06-15	2	Male	42.0	117
4	1030589	38578121	2019-05-26	2	Male	45.0	127
...	...	...	...	...	...	...	...
67	12574807	32056122	2019-05-25	2	Male	33.0	52
68	15192667	31272089	2019-05-24	2	Male	46.0	37
69	14248059	48790153	2019-05-21	2	Male	29.0	41
70	10629563	47864502	2019-06-01	2	Male	29.0	76
71	11737579	46626448	2019-05-27	2	Male	35.0	61

72 rows × 7 columns

In [133]:
```
test=pd.merge(data,data1)
```

In [134]:
```
test
```

Out[134]:

	customer_id	sex	customer_age	tenure	product_id	basket_date	basket_count
0	9500953	Male	55.0	96	3446783	2019-06-10	3
1	851739	Male	40.0	129	32920704	2019-06-19	2
2	9654043	Male	37.0	95	51307669	2019-06-08	2
3	4912369	Male	36.0	114	33923115	2019-05-20	2
4	9875271	Male	34.0	92	31586037	2019-06-06	2
...	...	...	...	...	...	...	...
67	13278573	Male	28.0	47	4488682	2019-05-26	2
68	12901520	Female	40.0	50	38610580	2019-05-28	3
69	12737235	Male	39.0	51	32933848	2019-05-21	2
70	12737235	Male	39.0	51	46373374	2019-05-21	3
71	12574807	Male	33.0	52	32056122	2019-05-25	2

72 rows × 7 columns

In [135]:
```
test.describe()
```

Out[135]:

	customer_id	customer_age	tenure	product_id	basket_count
count	7.200000e+01	72.000000	72.000000	7.200000e+01	72.000000
mean	1.554364e+07	68.458333	56.180556	3.140376e+07	2.152778
std	9.961282e+06	234.574289	38.948621	1.616160e+07	0.362298
min	3.809750e+05	5.000000	4.000000	8.287500e+04	2.000000
25%	1.026443e+07	29.000000	24.750000	2.980404e+07	2.000000
50%	1.352736e+07	35.500000	45.500000	3.498005e+07	2.000000
75%	2.037478e+07	43.000000	83.750000	4.359420e+07	2.000000
max	4.328080e+07	2022.000000	130.000000	5.130767e+07	3.000000

In [136]:
```
test.customer_id.unique()
```

Out[136]:
```
array([ 9500953,   851739,  9654043,  4912369,  9875271, 11737579,
       10619833,  4193819,  4897641,  4643359,   380975, 11623549,
       11724853, 12410433, 10394153,   537173, 11440499, 10439331,
       10629563,  4257099, 11346069,  8508353,  9700145, 10814041,
        9804585,  4238087, 11665521,  1030589, 11072047, 43280797,
       41790413, 39814593, 36623391, 34677755, 29144255, 27081691,
       25055107, 25567283, 23179191, 22524187, 21765975, 21142247,
       20789769, 20236456, 20174063, 17909829, 18256077, 17830393,
       16944627, 16398473, 16029475, 15436141, 15570891, 15192667,
       15067633, 14966315, 15141119, 14248059, 14053193, 13776147,
       13278573, 12901520, 12737235, 12574807])
```

In [137]:
```
data1.head()
```

Out[137]:

	customer_id	product_id	basket_date	basket_count
0	42366585	41475073	2019-06-19	2
1	35956841	43279538	2019-06-19	2
2	26139578	31715598	2019-06-19	3
3	3262253	47880260	2019-06-19	2
4	20056678	44747002	2019-06-19	2

In [138]:
```
data1.groupby(['product_id'])['basket_count'].sum().sort_values(ascending=False)
```

Out[138]:
```
product_id
43524799    69
31516269    59
39833031    50
46130148    36
34913531    28
            ..
34003520     2
34003697     2
34004660     2
34013459     2
55790974     2
Name: basket_count, Length: 13161, dtype: int64
```

In [139]:
```
data1.groupby(['product_id'])['basket_count'].sum().sort_values(ascending=True)
```

Out[139]:
```
product_id
49390        2
42094163     2
42102274     2
42110403     2
42110580     2
            ..
34913531    28
46130148    36
39833031    50
31516269    59
43524799    69
Name: basket_count, Length: 13161, dtype: int64
```

In [140]:
```
test.groupby(['customer_id']).count()
```

Out[140]:

customer_id	sex	customer_age	tenure	product_id	basket_date	basket_count
380975	2	2	2	2	2	2
537173	2	2	2	2	2	2
851739	1	1	1	1	1	1
1030589	1	1	1	1	1	1
4193819	1	1	1	1	1	1
...	...	...	...	...	...	...
34677755	1	1	1	1	1	1
36623391	1	1	1	1	1	1
39814593	2	2	2	2	2	2
41790413	1	1	1	1	1	1
43280797	1	1	1	1	1	1

64 rows × 6 columns

In [141]:
```
cor=data.corr()
cor
```

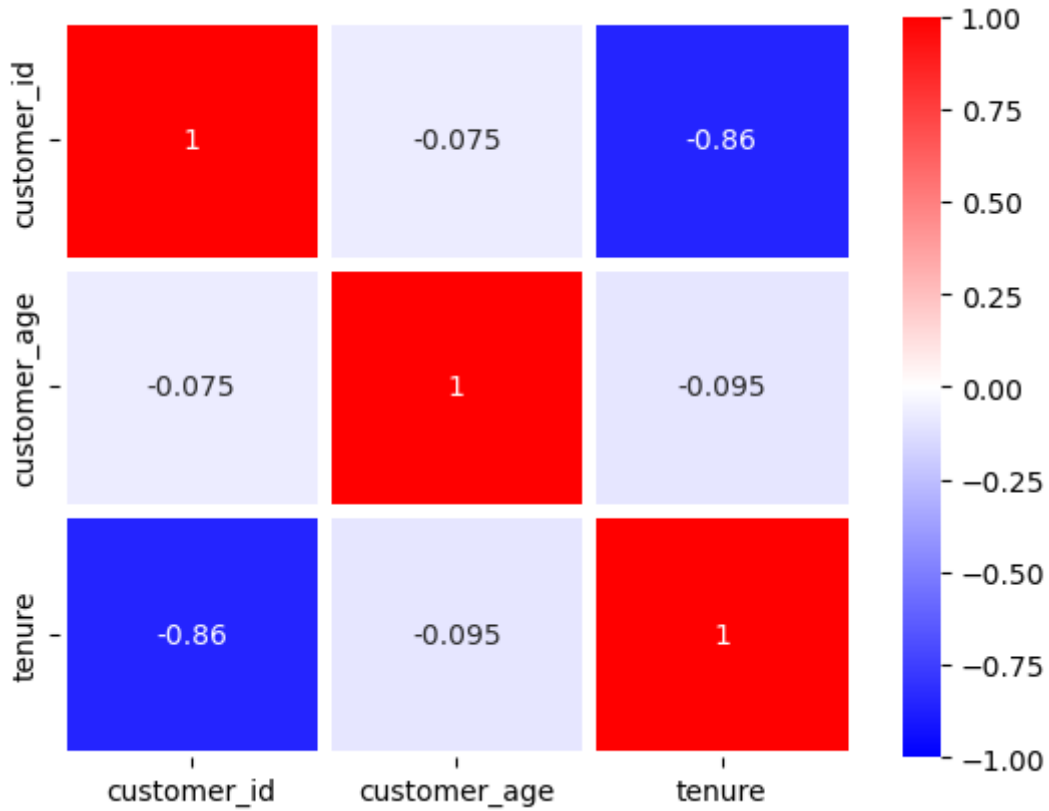
```
/tmp/ipykernel_5901/4173678507.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid columns or specify the value o
f numeric_only to silence this warning.
  cor=data.corr()
```

Out[141]:

	customer_id	customer_age	tenure
customer_id	1.000000	-0.075467	-0.855410
customer_age	-0.075467	1.000000	-0.095013
tenure	-0.855410	-0.095013	1.000000

In [142]:
```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='bwr')
```

Out[142]: <Axes: >



In [143]:
```
cor=data1.corr()
cor
```

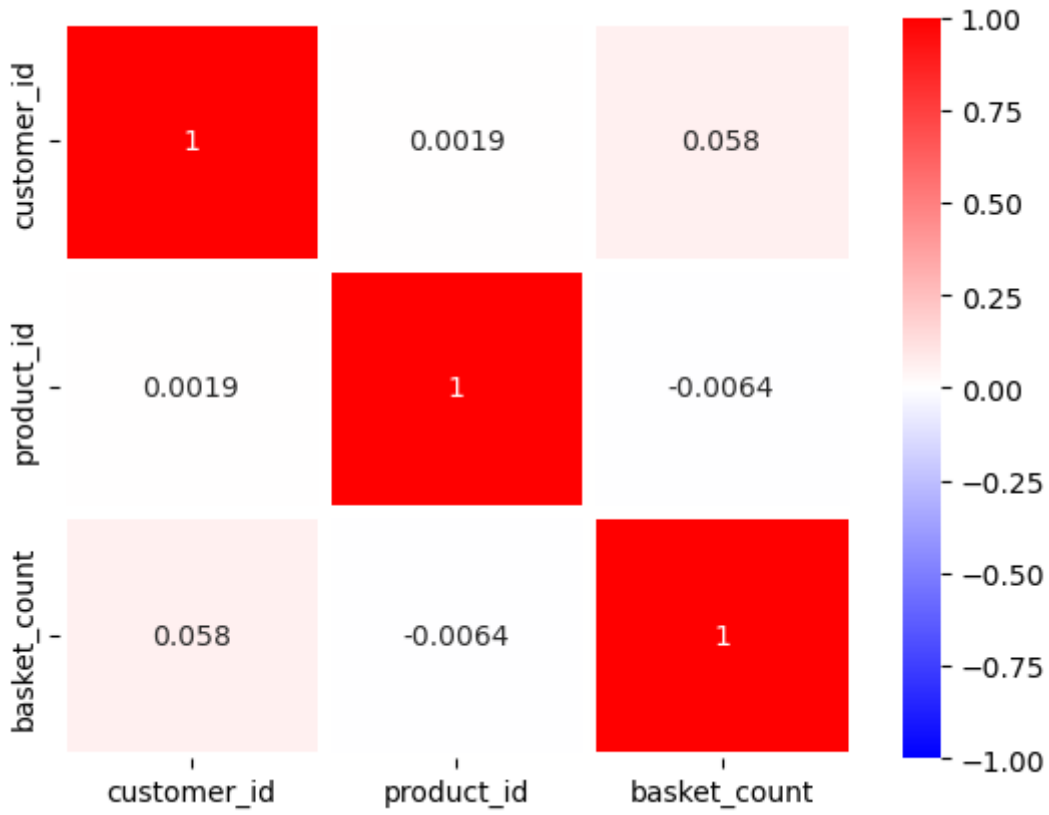
```
/tmp/ipykernel_5901/870474124.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is d
eprecated. In a future version, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  cor=data1.corr()
```

Out[143]:

	customer_id	product_id	basket_count
customer_id	1.000000	0.001937	0.058235
product_id	0.001937	1.000000	-0.006407
basket_count	0.058235	-0.006407	1.000000

In [144]:
```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='bwr')
```

Out[144]: <Axes: >



In [145]:
```
cor=test.corr()
cor
```

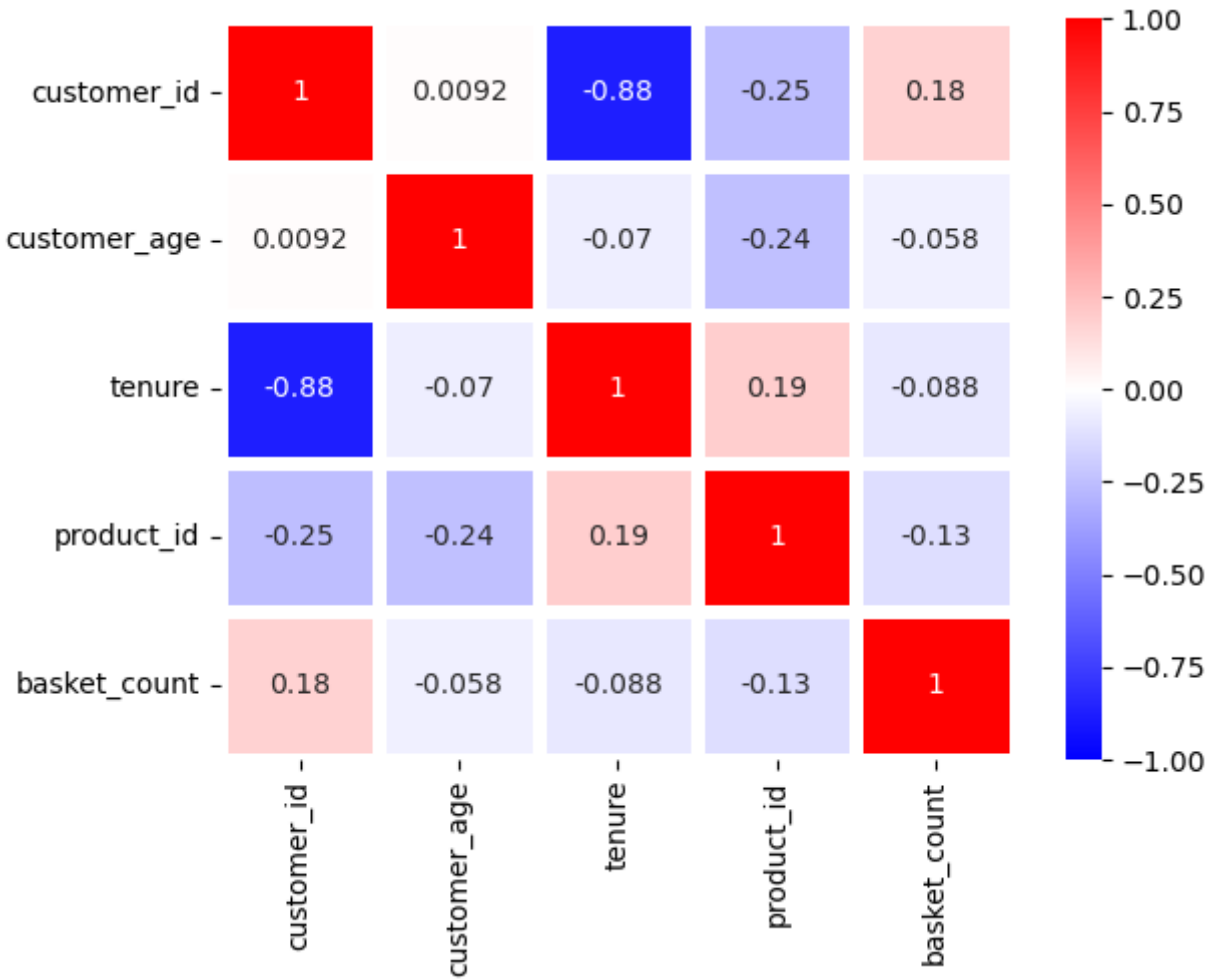
```
/tmp/ipykernel_5901/2206162927.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid columns or specify the value o
f numeric_only to silence this warning.
  cor=test.corr()
```

Out[145]:

	customer_id	customer_age	tenure	product_id	basket_count
customer_id	1.000000	0.009194	-0.882379	-0.252572	0.179558
customer_age	0.009194	1.000000	-0.069814	-0.243038	-0.058177
tenure	-0.882379	-0.069814	1.000000	0.190134	-0.087821
product_id	-0.252572	-0.243038	0.190134	1.000000	-0.125352
basket_count	0.179558	-0.058177	-0.087821	-0.125352	1.000000

In [146]:
```
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='bwr')
```

Out[146]: <Axes: >



In [ ]: