Data set 7

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\6_Salesworkload1.csv")
```

To print top rows:

In [8]:

```
b=a.head(1000)
b
```

Out[8]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLea
0	10.2016	1.0	United Kingdom	88253.0	London (I)	1.0	Dry	3184.764	0
1	10.2016	1.0	United Kingdom	88253.0	London (I)	2.0	Frozen	1582.941	0
2	10.2016	1.0	United Kingdom	88253.0	London (I)	3.0	other	47.205	0
3	10.2016	1.0	United Kingdom	88253.0	London (I)	4.0	Fish	1623.852	0
4	10.2016	1.0	United Kingdom	88253.0	London (I)	5.0	Fruits & Vegetables	1759.173	0
...	...	...	...	...	...	...	...	...	
995	11.2016	2.0	Poland	23623.0	Poznan	8.0	Household	1671.057	0
996	11.2016	2.0	Poland	23623.0	Poznan	9.0	Hardware	1516.854	0
997	11.2016	2.0	Poland	23623.0	Poznan	14.0	Non Food	5834.538	0
998	11.2016	2.0	Poland	23623.0	Poznan	15.0	Admin	3707.166	0
999	11.2016	2.0	Poland	23623.0	Poznan	12.0	Checkout	6312.882	0

1000 rows × 14 columns

To print Last rows:

In [4]:

```
a.tail()
```

Out[4]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	Hours
7653	06.2017	9.0	Sweden	29650.0	Gothenburg	12.0	Checkout	6322.323	
7654	06.2017	9.0	Sweden	29650.0	Gothenburg	16.0	Customer Services	4270.479	
7655	06.2017	9.0	Sweden	29650.0	Gothenburg	11.0	Delivery	0	
7656	06.2017	9.0	Sweden	29650.0	Gothenburg	17.0	others	2224.929	
7657	06.2017	9.0	Sweden	29650.0	Gothenburg	18.0	all	39652.2	

Statistical Summary:

In [5]:

```
a.describe()
```

Out[5]:

	Time index	StoreID	Dept_ID	HoursLease	Sales units	Turnover	Cu
count	7650.000000	7650.000000	7650.000000	7650.000000	7.650000e+03	7.650000e+03	
mean	5.000000	61995.220000	9.470588	22.036078	1.076471e+06	3.721393e+06	
std	2.582158	29924.581631	5.337429	133.299513	1.728113e+06	6.003380e+06	
min	1.000000	12227.000000	1.000000	0.000000	0.000000e+00	0.000000e+00	
25%	3.000000	29650.000000	5.000000	0.000000	5.457125e+04	2.726798e+05	
50%	5.000000	75400.500000	9.000000	0.000000	2.932300e+05	9.319575e+05	
75%	7.000000	87703.000000	14.000000	0.000000	9.175075e+05	3.264432e+06	
max	9.000000	98422.000000	18.000000	3984.000000	1.124296e+07	4.271739e+07	

To print no of rows and columns

In [6]:

```
a.shape
```

Out[6]:

```
(7658, 14)
```

To print no of elements

In [9]:

```
a.size
```

Out[9]:

```
107212
```

Missing no of values

In [10]:

```
a.isna()
```

Out[10]:

	MonthYear	Time index	Country	StoreID	City	Dept_ID	Dept. Name	HoursOwn	HoursLease	Sa u
0	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	
7653	False	False	False	False	False	False	False	False	False	F
7654	False	False	False	False	False	False	False	False	False	F
7655	False	False	False	False	False	False	False	False	False	F
7656	False	False	False	False	False	False	False	False	False	F
7657	False	False	False	False	False	False	False	False	False	F

7658 rows × 14 columns

In [14]:

```
a.dropna(axis=1,how='any')
```

Out[14]:

	MonthYear
0	10.2016
1	10.2016
2	10.2016
3	10.2016
4	10.2016
...	...
7653	06.2017
7654	06.2017
7655	06.2017
7656	06.2017
7657	06.2017

7658 rows × 1 columns

In [15]:

```
import matplotlib.pyplot as pp
```

In [17]:

```
c=b[['StoreID','HoursOwn','Area (m2)']]
c
```

Out[17]:

	StoreID	HoursOwn	Area (m2)
0	88253.0	3184.764	953.04
1	88253.0	1582.941	720.48
2	88253.0	47.205	966.72
3	88253.0	1623.852	1053.36
4	88253.0	1759.173	1053.36
...	...	...	...
995	23623.0	1671.057	4033.32
996	23623.0	1516.854	4158.72
997	23623.0	5834.538	14792.64
998	23623.0	3707.166	0
999	23623.0	6312.882	30684.24

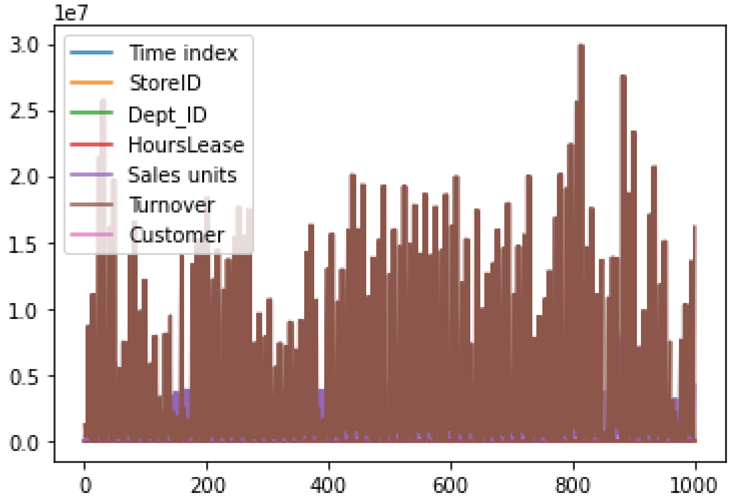
1000 rows × 3 columns

In [18]:

```
b.plot.line()
```

Out[18]:

```
<AxesSubplot:>
```

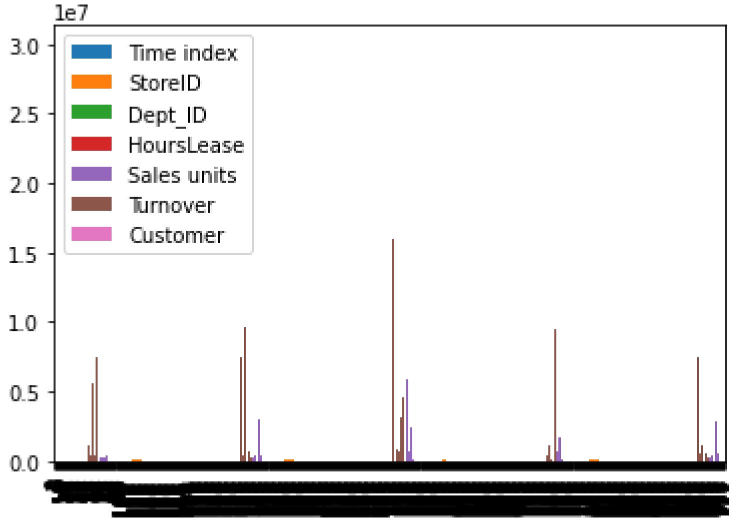


In [19]:

```
b.plot.bar()
```

Out[19]:

```
<AxesSubplot:>
```

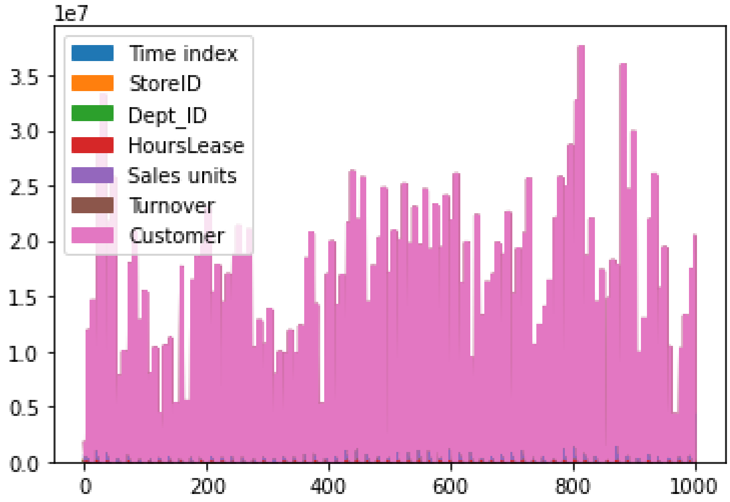


In [20]:

```
b.plot.area()
```

Out[20]:

```
<AxesSubplot:>
```

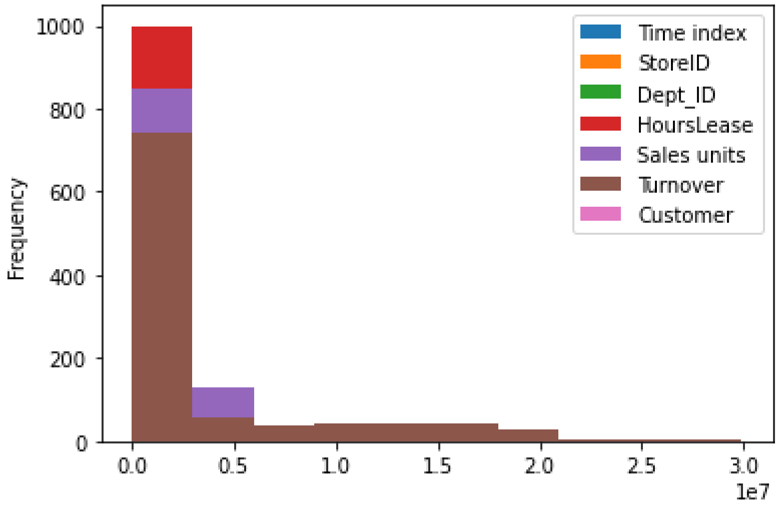


In [21]:

```
b.plot.hist()
```

Out[21]:

```
<AxesSubplot:ylabel='Frequency'>
```

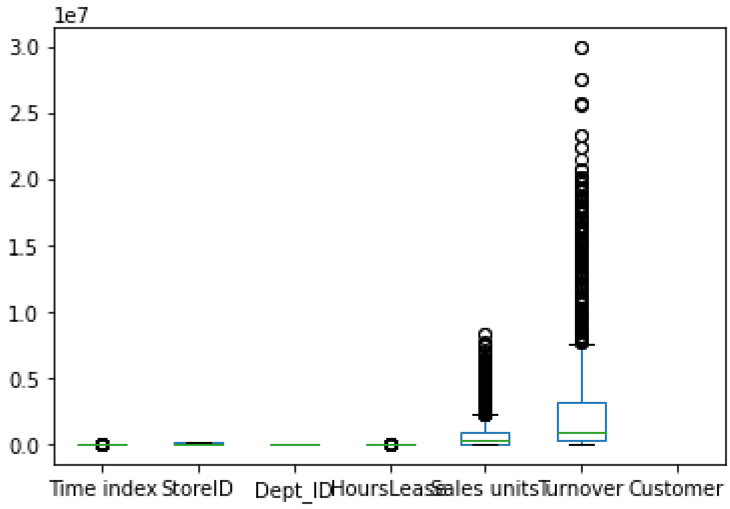


In [22]:

```
b.plot.box()
```

Out[22]:

```
<AxesSubplot:>
```

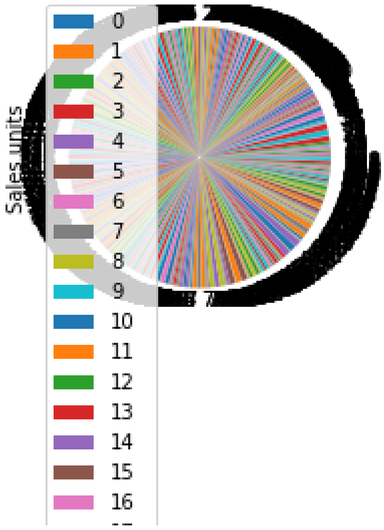


In [23]:

```
b.plot.pie(y='Sales units',figsize=(3,3))
```

Out[23]:

```
<AxesSubplot:ylabel='Sales units'>
```

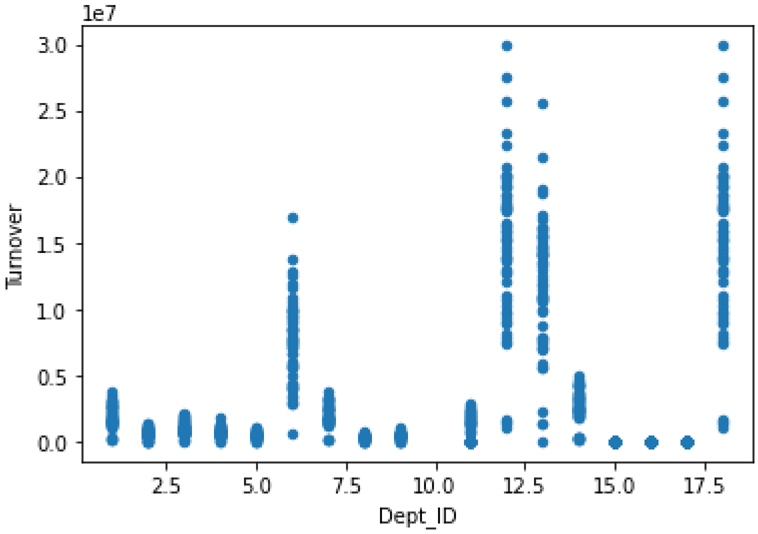


In [24]:

```
b.plot.scatter(x='Dept_ID',y='Turnover')
```

Out[24]:

```
<AxesSubplot:xlabel='Dept_ID', ylabel='Turnover'>
```



In [ ]:

```
```