In [20]:

```
# importing packages
import pandas as pd

# create data
df = pd.DataFrame([
                   [180000, 110, 18.9, 1400],
                   [360000, 905, 23.4, 1800],
                   [230000, 230, 14.0, 1300],
                   [60000, 450, 13.5, 1500]],

                   columns=['Col A', 'Col B',
                            'Col C', 'Col D'])

# view data
display(df)
```

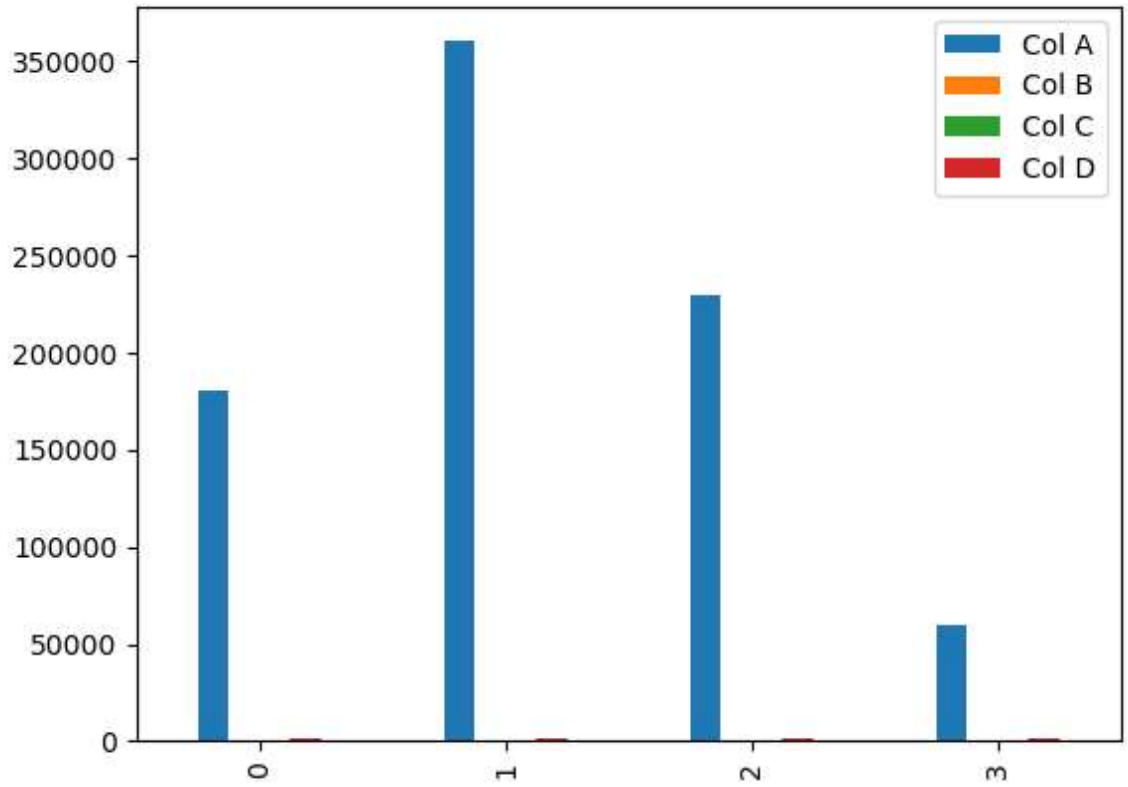
	Col A	Col B	Col C	Col D
0	180000	110	18.9	1400
1	360000	905	23.4	1800
2	230000	230	14.0	1300
3	60000	450	13.5	1500

In [21]:

```
import matplotlib.pyplot as plt
df.plot(kind = 'bar')
```

Out[21]:

<AxesSubplot:>



In [22]:

```
df.head(25)
```

Out[22]:

	Col A	Col B	Col C	Col D
0	180000	110	18.9	1400
1	360000	905	23.4	1800
2	230000	230	14.0	1300
3	60000	450	13.5	1500

In [23]:

```
df_max_scaled = df.copy()

# apply normalization techniques
for column in df_max_scaled.columns:
    df_max_scaled[column] = df_max_scaled[column]  / df_max_scaled[column].abs().max()

# view normalized data
display(df_max_scaled)
```

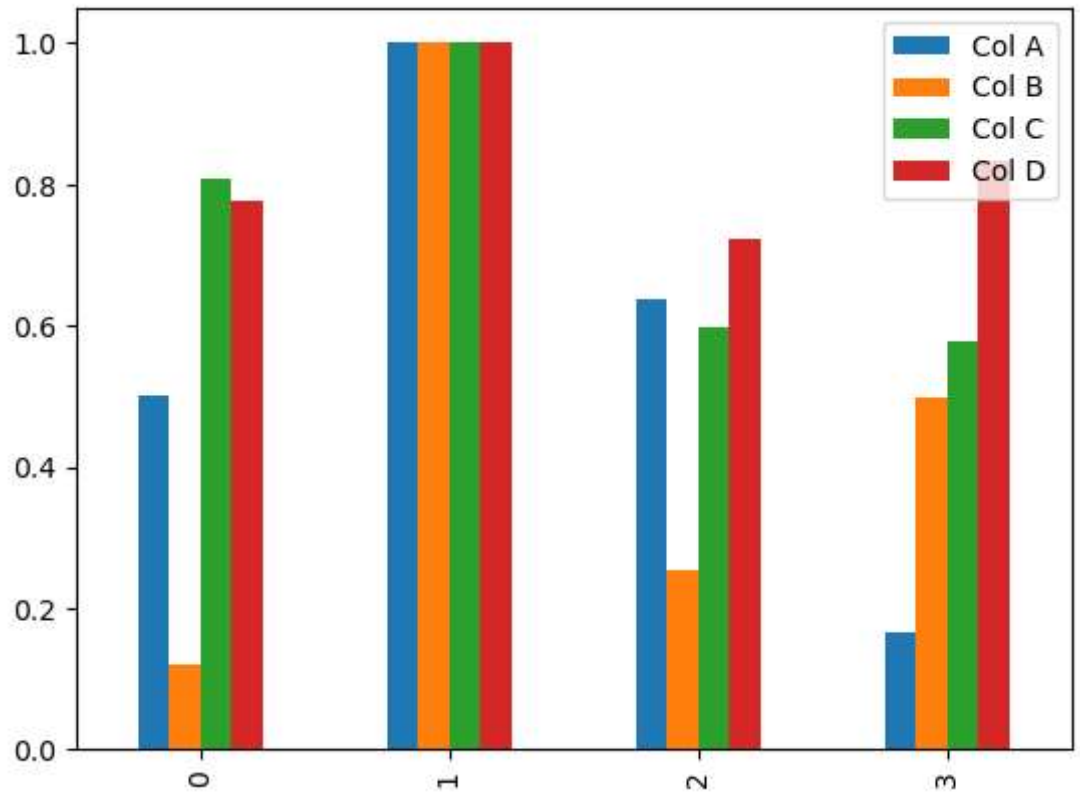
	Col A	Col B	Col C	Col D
0	0.500000	0.121547	0.807692	0.777778
1	1.000000	1.000000	1.000000	1.000000
2	0.638889	0.254144	0.598291	0.722222
3	0.166667	0.497238	0.576923	0.833333

In [24]:

```
import matplotlib.pyplot as plt
df_max_scaled.plot(kind = 'bar')
```

Out[24]:

<AxesSubplot:>



In [25]:

```
# copy the data
df_min_max_scaled = df.copy()

# apply normalization techniques
for column in df_min_max_scaled.columns:
    df_min_max_scaled[column] = (df_min_max_scaled[column] - df_min_max_scaled[column].m

# view normalized data
print(df_min_max_scaled)
```

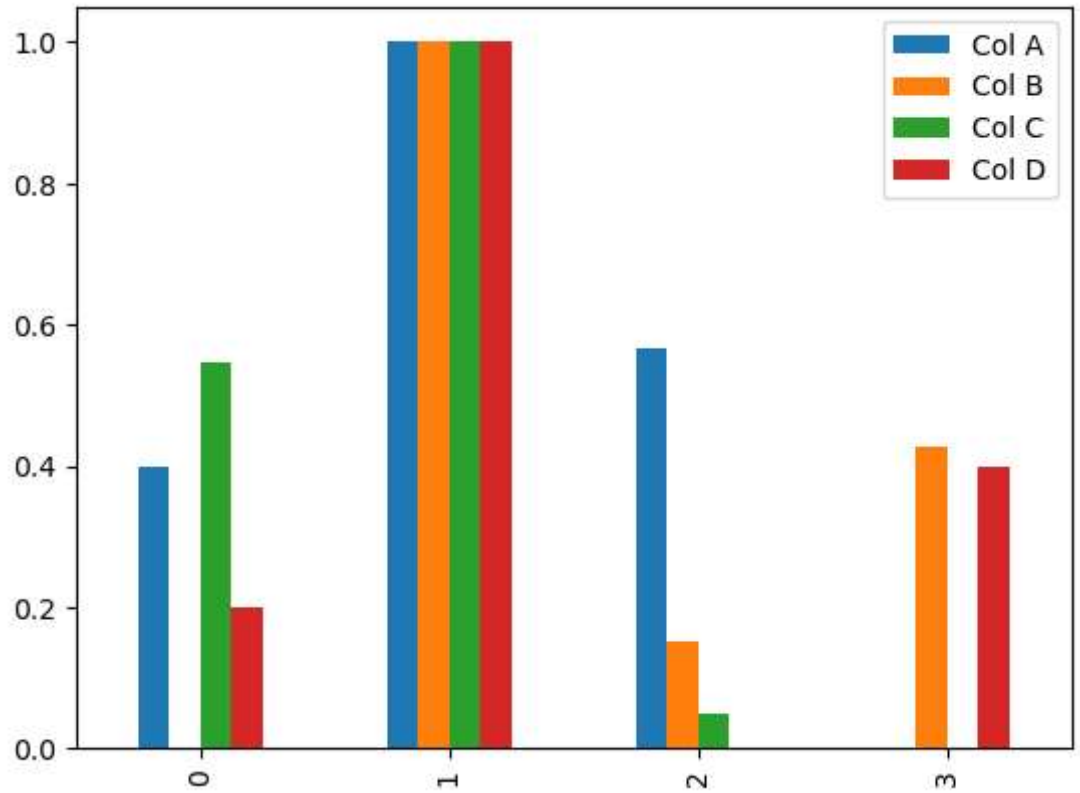
```
      Col A     Col B     Col C  Col D
0  0.400000  0.000000  0.545455    0.2
1  1.000000  1.000000  1.000000    1.0
2  0.566667  0.150943  0.050505    0.0
3  0.000000  0.427673  0.000000    0.4
```

In [26]:

```
import matplotlib.pyplot as plt
df_min_max_scaled.plot(kind = 'bar')
```

Out[26]:

```
<AxesSubplot:>
```



In [27]:

```
# copy the data
df_z_scaled = df.copy()

# apply normalization techniques
for column in df_z_scaled.columns:
    df_z_scaled[column] = (df_z_scaled[column] -
                           df_z_scaled[column].mean()) / df_z_scaled[column].std()

# view normalized data
display(df_z_scaled)
```

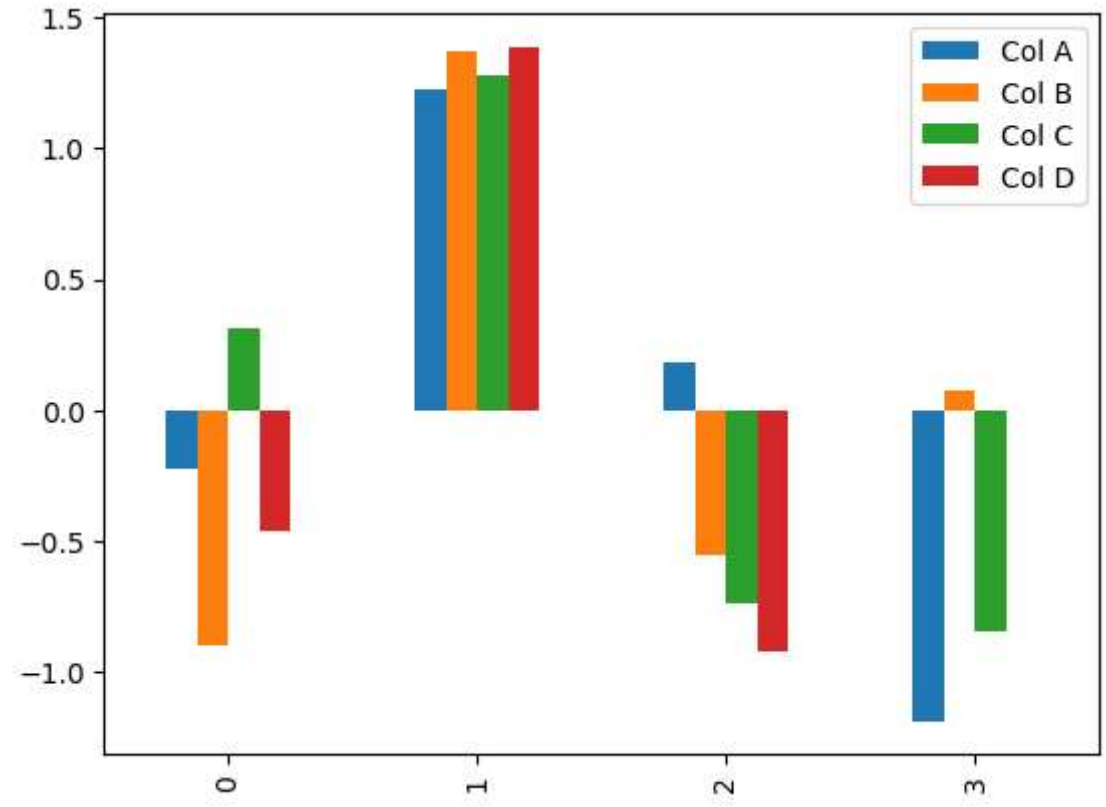
	Col A	Col B	Col C	Col D
0	-0.221422	-0.895492	0.311486	-0.46291
1	1.227884	1.373564	1.278167	1.38873
2	0.181163	-0.552993	-0.741122	-0.92582
3	-1.187625	0.074922	-0.848531	0.00000

In [28]:

```
import matplotlib.pyplot as plt
df_z_scaled.plot(kind='bar')
```

Out[28]:

```
<AxesSubplot:>
```



In [29]:

```
# standardization implementation
# Importing the library
import pandas as pd

# Creating the data frame
details = {
    'col1': [1, 3, 5, 7, 9],
    'col2': [7, 4, 35, 14, 56]
}

# creating a Dataframe object
df = pd.DataFrame(details)

# Z-Score using pandas
df['col1'] = (df['col1'] - df['col1'].mean()) / df['col1'].std()
```

In [30]:

```
# Importing the library
import pandas as pd
import scipy
from scipy import stats


# Creating the data frame
details = {
    'col1': [1, 3, 5, 7, 9],
    'col2': [7, 4, 35, 14, 56]
}

# creating a Dataframe object
df = pd.DataFrame(details)

# Z-Score using scipy
df['col2'] = stats.zscore(df['col2'])
```

In [31]:

```
# Importing the library
import pandas as pd
from sklearn.preprocessing import StandardScaler


# Creating the data frame
details = {
    'col1': [1, 3, 5, 7, 9],
    'col2': [7, 4, 35, 14, 56]
}

# creating a Dataframe object
df = pd.DataFrame(details)

# define standard scaler
scaler = StandardScaler()

# transform data
df = scaler.fit_transform(df)
```