We use Train.csv Dataset for this Internship

In [64]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [65]:
```
# loading the dataset using read_csv available in pandas
data = pd.read_csv("train.csv")
data
```

Out[65]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [66]:
```
#head- shows the first five entries in the data
data.head()
```

Out[66]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [67]:
```
# we gather Information about the data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [68]:
```
# getting sum of null values in the data
data.isnull().sum()
```

Out[68]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

TASK 1: PERFORMING DATA CLEANING

> Clean a dataset by removing missing values and outliers

cleaning the Dataset using fillna

In [69]:
```
# we replace the age null with mean or median of passengers
data['Age'].fillna(data['Age'].median())
```

Out[69]:
```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    28.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [70]:
```
# for cabins name that are empty simply you can drop them or fill with unkown value
data['Cabin'].fillna('Unknown')
```

Out[70]:
```
0      Unknown
1          C85
2      Unknown
3         C123
4      Unknown
        ...
886    Unknown
887        B42
888    Unknown
889       C148
890    Unknown
Name: Cabin, Length: 891, dtype: object
```

In [71]:
```
# for Embarked Attribute We use Most Frequently repeated value using mode() function
data['Embarked'].fillna(data['Embarked'].mode()[0])
```

Out[71]:
```
0      S
1      C
2      S
3      S
4      S
      ..
886    S
887    S
888    S
889    C
890    Q
Name: Embarked, Length: 891, dtype: object
```

In [44]:
```
# verification if all the missing values are removed or not !
data.isnull().sum()
```

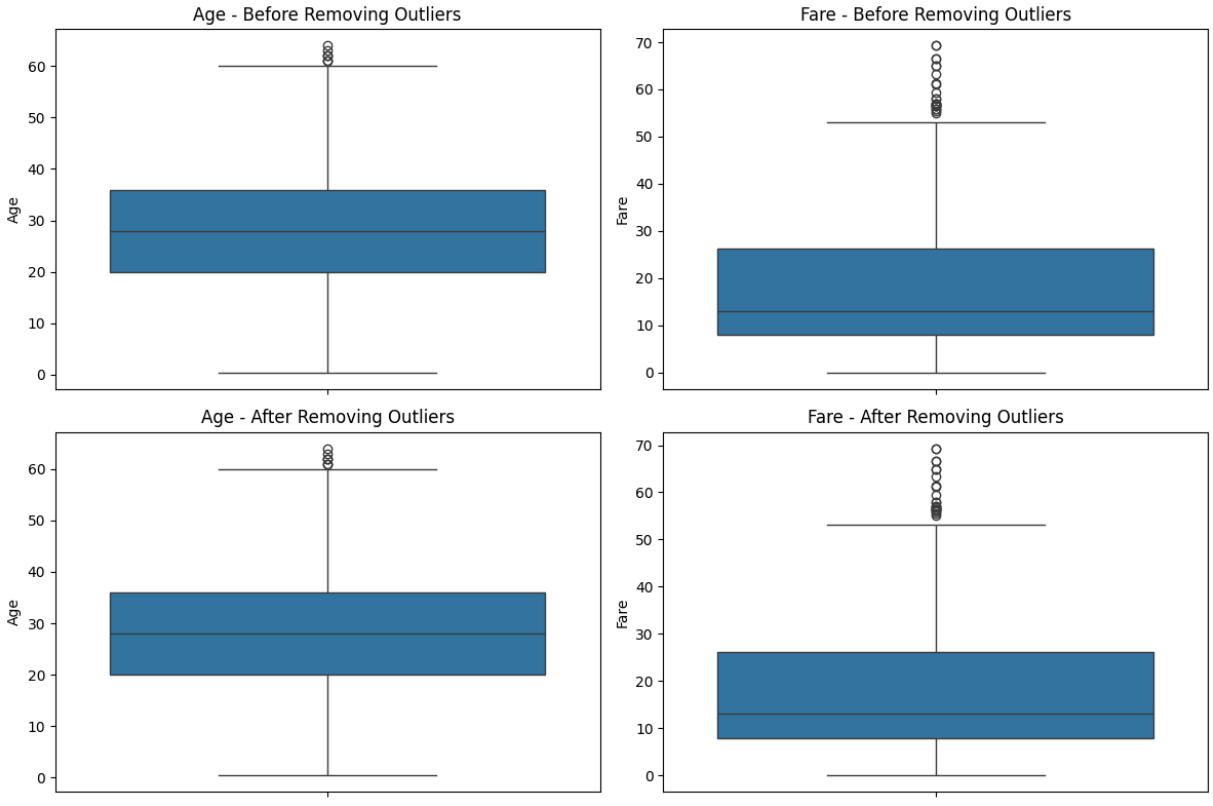
Out[44]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

Handling Outliers

In [72]:
```
# Imagine data as a line of numbers. Outliers are numbers that don't fit with the rest. To clean them:
# For this we keep middle aged people (from 25 - 50 years) and remove old aged and children
# Similar to age we keep middle fares and remove very cheap and high Fares
```

In [73]:
```
# We use function for removing the outliers
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q2 = df[column].quantile(0.75)
    IQR = Q2 - Q1
    low_bnd = Q1 - 1.5 * IQR
    up_bnd = Q2 + 1.5 * IQR
    return df[(df[column] >= low_bnd) & (df[column] <= up_bnd)]
data = remove_outliers(data, 'Age')
data = remove_outliers(data, 'Fare')
data.to_csv('clean.csv', index = False)
```

In [74]:
```
import seaborn as sns
clean = pd.read_csv('clean.csv')
plt.figure(figsize=(12, 8))
plt.subplot(2, 2, 1)
sns.boxplot(y=data['Age'])
plt.title('Age - Before Removing Outliers')
plt.subplot(2, 2, 2)
sns.boxplot(y=data['Fare'])
plt.title('Fare - Before Removing Outliers')
plt.subplot(2, 2, 3)
sns.boxplot(y=clean['Age'])
plt.title('Age - After Removing Outliers')
plt.subplot(2, 2, 4)
sns.boxplot(y=clean['Fare'])
plt.title('Fare - After Removing Outliers')
plt.tight_layout()
plt.show()
```



In [75]:
```
clean = pd.read_csv('clean.csv')
print("\nSummary before Removing Outliers")
print(data[['Age', 'Fare']].describe())
print("\nSummary After Removing Outliers")
print(clean[['Age', 'Fare']].describe())
```

```
Summary before Removing Outliers
              Age        Fare
count  607.000000  607.000000
mean    28.246705   18.933401
std     13.429036   14.184362
min      0.420000    0.000000
25%     20.000000    7.925000
50%     28.000000   13.000000
75%     36.000000   26.250000
max     64.000000   69.300000

Summary After Removing Outliers
              Age        Fare
count  607.000000  607.000000
mean    28.246705   18.933401
std     13.429036   14.184362
min      0.420000    0.000000
25%     20.000000    7.925000
50%     28.000000   13.000000
75%     36.000000   26.250000
max     64.000000   69.300000
```

In [ ]: