In [1]:
```
!pip install mysql.connector
```

```
Requirement already satisfied: mysql.connector in c:\users\user15\anaconda3\lib\site-packages (2.2.9)
```

In [2]:
```
!pip install pymysql
```

```
Requirement already satisfied: pymysql in c:\users\user15\anaconda3\lib\site-packages (1.0.2)
```

In [3]:
```
import pymysql
conn = pymysql.connect(host = 'localhost', user = 'root',password = 'root', database = 'aarti')
```

In [4]:
```
import pandas as pd
df = pd.read_sql_query('select * from student',conn)
df
```

```
C:\Users\User15\anaconda3\lib\site-packages\pandas\io\sql.py:761: UserWarning: pandas only support SQLAlchemy connectable(engin
e/connection) ordatabase string URI or sqlite3 DBAPI2 connectionother DBAPI2 objects are not tested, please consider using SQLA
lchemy
  warnings.warn(
```

Out[4]:

	roll_no	name	course_name
0	1	aarti	Msc DSAI
1	2	abhi	Msc DSAI
2	3	pranoti	Msc it
3	4	anushka	Msc cs
4	5	pooja	Msc it
5	6	swapnil	Msc cs

In [5]:
```
df = pd.read_csv('datasets/train - train.csv')
df
```

Out[5]:

	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

In [6]:
```
# How Big is your dataset
df.shape
```

Out[6]: (891, 12)

In [7]:
```
df.sample()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
708	709	1	1	Cleaver, Miss. Alice	female	22.0	0	0	113781	151.55	NaN	S

In [8]:
```
# show mean std max and min and max
df.describe()
```

Out[8]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [9]:
```
df.dtypes
```

Out[9]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [10]:
```
#
df.isnull().sum()
```

Out[10]:
```
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
```

In [11]:
```
# 6. Is there any duplicate values ?
df.duplicated().sum()
```

Out[11]:
```
0
```

In [12]:
```
# 7. Display correlation of all columns with survived column.

df.corr()['Survived']
```

Out[12]:
```
PassengerId   -0.005007
Survived       1.000000
Pclass        -0.338481
Age           -0.077221
SibSp         -0.035322
Parch          0.081629
Fare           0.257307
Name: Survived, dtype: float64
```

In [ ]:
```

```