Importing necessary modules

In [23]:
```
import pandas as pd
import mysql.connector as sql
from sqlalchemy import create_engine
engine = create_engine("mysql+pymysql://{user}:{pw}@{host}/{db}".format(host="localhost", db='assign', user="root", pw="praveen123"))
import mysql.connector as server1
db1=server1.connect(host='localhost',user='root',password='praveen123',database='assign')
mycursor=db1.cursor()
```

In [24]:
```
house=pd.read_csv(r"K:\Desktop\NIIT\tables\DS1_C4_S6_House_Prices_Data_Challenge.csv")
house
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208500
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181500
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223500
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140000
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	8	2007	WD	Normal	175000
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD	Normal	210000
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	5	2010	WD	Normal	266500
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	2010	WD	Normal	142125
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2008	WD	Normal	147500

1460 rows × 81 columns

Task1: The most people who prefer houses more than 6000ft and want a corner homes

In [25]:
```
grand_6k=house[(house.LotConfig=="Corner")&(house.LotArea>6000)]
```

In [26]:
```
grand_6k.to_sql( name='task1',con=engine,if_exists='append',index=False)
```

Out[26]: 238

Task2: To find people who most prefer preferably on land and one family and one storey

In [27]:
```
task2=house[(house.BldgType=="1Fam")&(house.HouseStyle=="1Story")&(house.LandContour=="Lvl")]
```

In [28]:
```
task2.to_sql( name='task2',con=engine,if_exists='append',index=False)
```

Out[28]: 545

Task3: To find compact townhouses which are hot selling

In [29]:
```
Task3=house[(house.BldgType=="TwnhsE")|(house.BldgType=="Twnhs")]
```

In [30]:
```
Task3.to_sql( name='task3',con=engine,if_exists='append',index=False)
```

Out[30]: 157

Task4: To find house who have basement 3+ bedrooms and cement exteriors

In [31]:
```
task4=house[(house.BsmtQual!="NaN")&(house.Exterior1st=="CemntBd")&(house.TotRmsAbvGrd>3)]
```

In [32]:
```
task4.to_sql( name='task4',con=engine,if_exists='append',index=False)
```

Out[32]: 55

Task5: To find garage area more than 500+ sqft

In [33]:
```
task5=house[house.GarageArea>500]
```

In [34]:
```
task5.to_sql( name='task5',con=engine,if_exists='append',index=False)
```

Out[34]: 629

In [35]:
```
dis="""SELECT table_name
FROM information_schema.tables
WHERE table_type='BASE TABLE'"""
mycursor.execute(dis)

print("The list of all tables inside the dataframe")
tables=mycursor.fetchall()
tables[:5]
```

```
The list of all tables inside the dataframe
```

Out[35]: [('task1',), ('task2',), ('task3',), ('task4',), ('task5',)]