In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns

# Import label encoder
from sklearn import preprocessing


import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

Importing the Dataset

In [31]:

```
#IMPORTING DATASET
df = pd.read_csv('PEP1.csv')
df.head()
```

Out[31]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl

5 rows × 81 columns

Understanding the Dataset

In [3]:

```
df.shape
```

Out[3]:

```
(1460, 81)
```

In [4]:

```
df.isna()
```

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandCo
0	False	False	False	False	False	False	True	False	
1	False	False	False	False	False	False	True	False	
2	False	False	False	False	False	False	True	False	
3	False	False	False	False	False	False	True	False	
4	False	False	False	False	False	False	True	False	
...	...	...	...	...	...	...	...	...	
1455	False	False	False	False	False	False	True	False	
1456	False	False	False	False	False	False	True	False	
1457	False	False	False	False	False	False	True	False	
1458	False	False	False	False	False	False	True	False	
1459	False	False	False	False	False	False	True	False	

1460 rows × 81 columns

In [5]:

```
df.isna().sum(axis=0)
```

Out[5]:

```
Id                 0
MSSubClass         0
MSZoning           0
LotFrontage      259
LotArea            0
                ...
MoSold             0
YrSold             0
SaleType           0
SaleCondition      0
SalePrice          0
Length: 81, dtype: int64
```

EDA of Numerical Variables

In [6]:

```
#UnIQUE
numeric_df = df.select_dtypes(include=[np.number])
numericcol = numeric_df.columns.to_list()

#print("\n Numeric :",numeric_df)
print("\n Numeric :",numericcol)
```

```
 Numeric : ['Id', 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual',
'OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'B
smtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFi
nSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
'BedroomAbvGr', 'KitchebvGr', 'TotRmsAbvGrd', 'Fireplaces', 'GarageYrBlt',
'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch',
'3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'Sa
lePrice']
```

In [7]:

```
#Missing values of numerical
numeric_df.isna()
```

Out[7]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRe
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
1455	False	False	False	False	False	False	False	
1456	False	False	False	False	False	False	False	
1457	False	False	False	False	False	False	False	
1458	False	False	False	False	False	False	False	
1459	False	False	False	False	False	False	False	

1460 rows × 38 columns

In [8]:

```
numeric_df.isna().sum(axis=0)
```

Out[8]:

```
Id                 0
MSSubClass         0
LotFrontage      259
LotArea            0
OverallQual        0
OverallCond        0
YearBuilt          0
YearRemodAdd       0
MasVnrArea         8
BsmtFinSF1         0
BsmtFinSF2         0
BsmtUnfSF          0
TotalBsmtSF        0
1stFlrSF           0
2ndFlrSF           0
LowQualFinSF       0
GrLivArea          0
BsmtFullBath       0
BsmtHalfBath       0
FullBath           0
HalfBath           0
BedroomAbvGr       0
KitchebvGr         0
TotRmsAbvGrd       0
Fireplaces         0
GarageYrBlt       81
GarageCars         0
GarageArea         0
WoodDeckSF         0
OpenPorchSF        0
EnclosedPorch      0
3SsnPorch          0
ScreenPorch        0
PoolArea           0
MiscVal            0
MoSold             0
YrSold             0
SalePrice          0
dtype: int64
```

In [9]:

```
#percentage of missing value
per_missing_value = (numeric_df['LotFrontage'].isna().sum(axis=0)/df.shape[0])*100
per_missing_value
```

Out[9]:

```
17.73972602739726
```

In [10]:

```
numeric_df_r=numeric_df.dropna()
numeric_df_r
```

Out[10]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRer
0	1	60	65.0	8450	7	5	2003	
1	2	20	80.0	9600	6	8	1976	
2	3	60	68.0	11250	7	5	2001	
3	4	70	60.0	9550	7	5	1915	
4	5	60	84.0	14260	8	5	2000	
...	...	...	...	...	...	...	...	
1455	1456	60	62.0	7917	6	5	1999	
1456	1457	20	85.0	13175	6	6	1978	
1457	1458	70	66.0	9042	7	9	1941	
1458	1459	20	68.0	9717	5	6	1950	
1459	1460	20	75.0	9937	5	6	1965	

1121 rows × 38 columns

In [11]:

```
numeric_df_rc=numeric_df_r.dropna(axis='columns')
numeric_df_rc
```

Out[11]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRer
0	1	60	65.0	8450	7	5	2003	
1	2	20	80.0	9600	6	8	1976	
2	3	60	68.0	11250	7	5	2001	
3	4	70	60.0	9550	7	5	1915	
4	5	60	84.0	14260	8	5	2000	
...	...	...	...	...	...	...	...	
1455	1456	60	62.0	7917	6	5	1999	
1456	1457	20	85.0	13175	6	6	1978	
1457	1458	70	66.0	9042	7	9	1941	
1458	1459	20	68.0	9717	5	6	1950	
1459	1460	20	75.0	9937	5	6	1965	

1121 rows × 38 columns

In [12]:

```
numeric_df_rc.isna().sum(axis=0)
```

Out[12]:

```
Id               0
MSSubClass       0
LotFrontage      0
LotArea          0
OverallQual      0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
MasVnrArea       0
BsmtFinSF1       0
BsmtFinSF2       0
BsmtUnfSF        0
TotalBsmtSF      0
1stFlrSF         0
2ndFlrSF         0
LowQualFinSF     0
GrLivArea        0
BsmtFullBath     0
BsmtHalfBath     0
FullBath         0
HalfBath         0
BedroomAbvGr     0
KitchebvGr       0
TotRmsAbvGrd     0
Fireplaces       0
GarageYrBlt      0
GarageCars       0
GarageArea       0
WoodDeckSF       0
OpenPorchSF      0
EnclosedPorch    0
3SsnPorch        0
ScreenPorch      0
PoolArea         0
MiscVal          0
MoSold           0
YrSold           0
SalePrice        0
dtype: int64
```

In [13]:

```
#Droping column with Missing value
for i in numeric_df_rc.columns:
    if numeric_df_rc[i].isnull().count()>0:
        df= numeric_df_rc.drop(i,axis=1)
```

In [14]:

```
df.shape
```

Out[14]:

```
(1121, 37)
```

In [15]:

```
#Removing rows with missing values
numeric_df_rcl=numeric_df_rc.loc[:, ~numeric_df.isnull().any(axis=0)]
numeric_df_rcl
```

Out[15]:

	Id	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	Bsm
0	1	60	8450	7	5	2003	2003	
1	2	20	9600	6	8	1976	1976	
2	3	60	11250	7	5	2001	2002	
3	4	70	9550	7	5	1915	1970	
4	5	60	14260	8	5	2000	2000	
...	...	...	...	...	...	...	...	
1455	1456	60	7917	6	5	1999	2000	
1456	1457	20	13175	6	6	1978	1988	
1457	1458	70	9042	7	9	1941	2006	
1458	1459	20	9717	5	6	1950	1996	
1459	1460	20	9937	5	6	1965	1965	

1121 rows × 35 columns

In [16]:

```
#ChECKING SKEWNESS
numeric_df_rcl.skew(axis=0,skipna=True)
```

Out[16]:

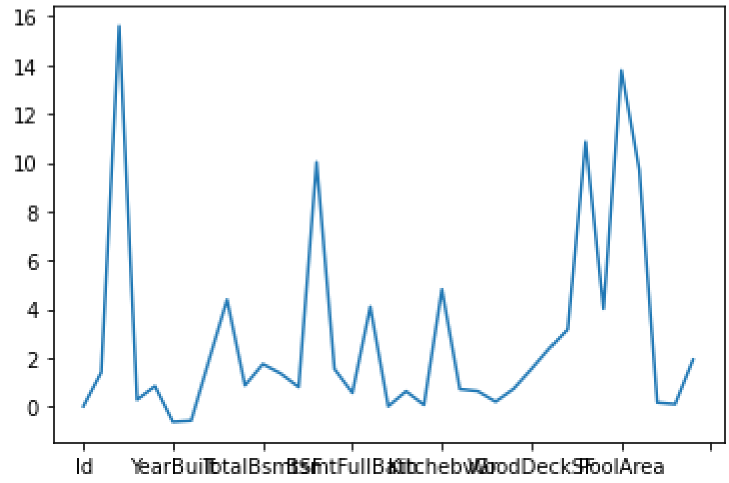
```
Id               0.018663
MSSubClass       1.412907
LotArea         15.608113
OverallQual      0.287800
OverallCond      0.846451
YearBuilt       -0.618350
YearRemodAdd    -0.565757
BsmtFinSF1       1.934077
BsmtFinSF2       4.399358
BsmtUnfSF        0.875774
TotalBsmtSF      1.754916
1stFlrSF         1.363783
2ndFlrSF         0.807411
LowQualFinSF    10.020823
GrLivArea        1.549961
BsmtFullBath     0.568804
BsmtHalfBath     4.107874
FullBath         0.015822
HalfBath         0.638178
BedroomAbvGr     0.074427
KitchebvGr       4.822542
TotRmsAbvGrd     0.723117
Fireplaces       0.643698
GarageCars       0.206017
GarageArea       0.733894
WoodDeckSF       1.549793
OpenPorchSF      2.403928
EnclosedPorch    3.173250
3SsnPorch       10.854868
ScreenPorch      4.019111
PoolArea        13.783823
MiscVal          9.699989
MoSold           0.173039
YrSold           0.106730
SalePrice        1.933615
dtype: float64
```

In [17]:

```
numeric_df_rcl.skew(axis=0,skipna=True).plot()
```

Out[17]:

<AxesSubplot:>

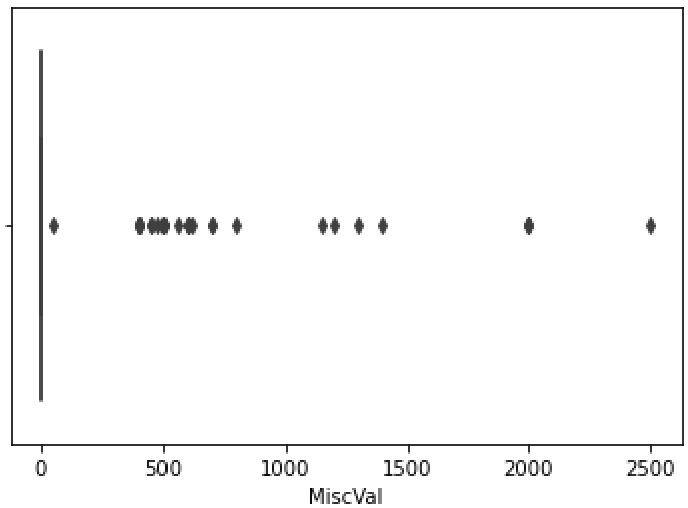


In [18]:

```
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

sns.boxplot(x=numeric_df_rcl['MiscVal'])
```

Out[18]:

<AxesSubplot:xlabel='MiscVal'>



In [19]:

```
#ChECKING SKEWNESS
# Importing numpy and statsmodels
import numpy as np
from statsmodels.stats.stattools import medcouple
from statsmodels.stats.stattools import robust_skewness

x = np.array(numeric_df_rcl['MiscVal'])
# Using statsmodels.robust_skewness() method
skewness = medcouple(x)

print(skewness)
```

```
1.0
```

In [20]:

```
#Correleation

pearsoncorr = numeric_df_rcl.corr(method='pearson')
pearsoncorr
```

Out[20]:

	Id	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	Year
Id	1.000000	0.021937	-0.040711	-0.058269	0.004387	-0.020862	
MSSubClass	0.021937	1.000000	-0.198096	0.029522	-0.087859	0.025800	
LotArea	-0.040711	-0.198096	1.000000	0.167525	-0.034348	0.029205	
OverallQual	-0.058269	0.029522	0.167525	1.000000	-0.163157	0.589385	
OverallCond	0.004387	-0.087859	-0.034348	-0.163157	1.000000	-0.426462	
YearBuilt	-0.020862	0.025800	0.029205	0.589385	-0.426462	1.000000	
YearRemodAdd	-0.027664	0.006645	0.026848	0.570757	0.039402	0.623171	
BsmtFinSF1	-0.013751	-0.070389	0.230441	0.249500	-0.054788	0.236941	
BsmtFinSF2	0.012544	-0.075439	0.138234	-0.068506	0.042314	-0.054414	
BsmtUnfSF	-0.012985	-0.145582	0.011288	0.322663	-0.148630	0.177545	
TotalBsmtSF	-0.023129	-0.247781	0.302554	0.563960	-0.192762	0.409134	
1stFlrSF	-0.008046	-0.252249	0.329679	0.514453	-0.164251	0.308875	
2ndFlrSF	-0.002346	0.319328	0.074612	0.273197	0.005985	-0.011621	
LowQualFinSF	-0.039933	0.024704	0.020039	-0.008118	0.048720	-0.164359	
GrLivArea	-0.011068	0.083365	0.307164	0.607466	-0.112231	0.204967	
BsmtFullBath	0.026113	-0.014681	0.179052	0.126834	-0.060943	0.182800	
BsmtHalfBath	-0.026774	0.012310	-0.014282	-0.053283	0.122960	-0.049645	
FullBath	0.007220	0.131278	0.129073	0.576875	-0.229848	0.500495	
HalfBath	-0.010409	0.203971	0.045183	0.251690	-0.079023	0.220000	
BedroomAbvGr	0.039831	-0.032971	0.137269	0.094882	0.004643	-0.061580	
KitchebvGr	0.025913	0.266012	-0.018942	-0.178735	-0.092644	-0.171920	
TotRmsAbvGrd	0.020012	0.047209	0.237918	0.451008	-0.096901	0.121417	
Fireplaces	-0.018273	-0.031122	0.255755	0.415294	-0.022290	0.133077	
GarageCars	-0.008125	-0.027411	0.172428	0.593803	-0.267859	0.532563	
GarageArea	-0.025889	-0.092607	0.211362	0.550659	-0.226347	0.471286	
WoodDeckSF	-0.025060	-0.017988	0.133576	0.282512	-0.010835	0.238548	
OpenPorchSF	-0.001972	0.004054	0.099170	0.340679	-0.076273	0.235432	
EnclosedPorch	0.009935	-0.017790	-0.023631	-0.144344	0.062748	-0.392693	
3SsnPorch	-0.066833	-0.039739	0.012520	0.017331	-0.006861	0.027948	
ScreenPorch	0.015183	-0.021789	0.072517	0.055296	0.087030	-0.063694	
PoolArea	0.048010	0.003166	0.109147	0.080131	-0.023566	0.006717	
MiscVal	0.045799	-0.040689	0.012790	-0.062064	0.119772	-0.096973	
MoSold	-0.000570	-0.027170	0.008998	0.079895	-0.014236	0.013784	
YrSold	0.013407	-0.012448	-0.006904	-0.008903	0.041003	-0.004585	
SalePrice	-0.047122	-0.088032	0.299962	0.797881	-0.124391	0.525394	

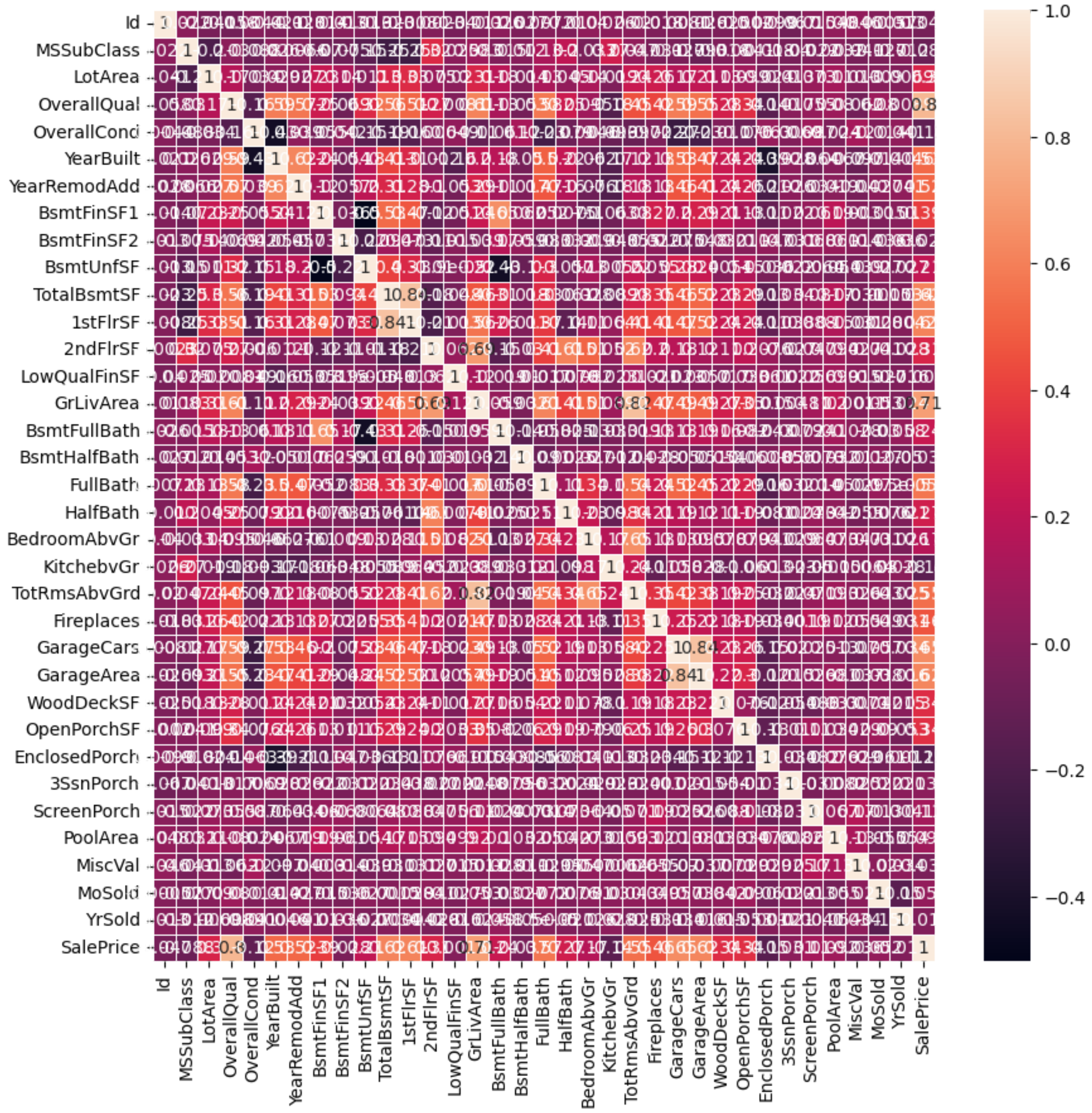
35 rows × 35 columns

In [21]:

```
#corrmatrix

plt.figure(figsize=(10,10), dpi=100)
sns.heatmap(pearsoncorr,
xticklabels=pearsoncorr.columns,
yticklabels=pearsoncorr.columns,
annot=True,
linewidths=0.5)
```

Out[21]:

```
<AxesSubplot:>
```



In [22]:

```
#Correlation with output variables
cor_target = abs(pearsoncorr['SalePrice'])
cor_target
```

Out[22]:

```
Id               0.047122
MSSubClass       0.088032
LotArea          0.299962
OverallQual      0.797881
OverallCond      0.124391
YearBuilt        0.525394
YearRemodAdd     0.521253
BsmtFinSF1       0.390301
BsmtFinSF2       0.028021
BsmtUnfSF        0.213129
TotalBsmtSF      0.615612
1stFlrSF         0.607969
2ndFlrSF         0.306879
LowQualFinSF     0.001482
GrLivArea        0.705154
BsmtFullBath     0.236737
BsmtHalfBath     0.036513
FullBath         0.566627
HalfBath         0.268560
BedroomAbvGr     0.166814
KitchebvGr       0.140497
TotRmsAbvGrd     0.547067
Fireplaces       0.461873
GarageCars       0.647034
GarageArea       0.619330
WoodDeckSF       0.336855
OpenPorchSF      0.343354
EnclosedPorch    0.154843
3SsnPorch        0.030777
ScreenPorch      0.110427
PoolArea         0.092488
MiscVal          0.036041
MoSold           0.051568
YrSold           0.011869
SalePrice        1.000000
Name: SalePrice, dtype: float64
```

In [23]:

```
relevant_feaures = cor_target[cor_target>0.5]
relevant_feaures
```

Out[23]:

```
OverallQual     0.797881
YearBuilt       0.525394
YearRemodAdd    0.521253
TotalBsmtSF     0.615612
1stFlrSF        0.607969
GrLivArea       0.705154
FullBath        0.566627
TotRmsAbvGrd    0.547067
GarageCars      0.647034
GarageArea      0.619330
SalePrice       1.000000
Name: SalePrice, dtype: float64
```

In [ ]:

In [24]:

```
relevant_feaures_df = numeric_df_rc[['OverallQual','YearBuilt','YearRemodAdd','TotalBsmtS
```

In [25]:

```
relevant_feaures_df.shape
```

Out[25]:

```
(1121, 11)
```

In [26]:

```
#RELEVANT FEATURES OF NUMERICAL VARIABLES
relevant_feaures_df.head()
```

Out[26]:

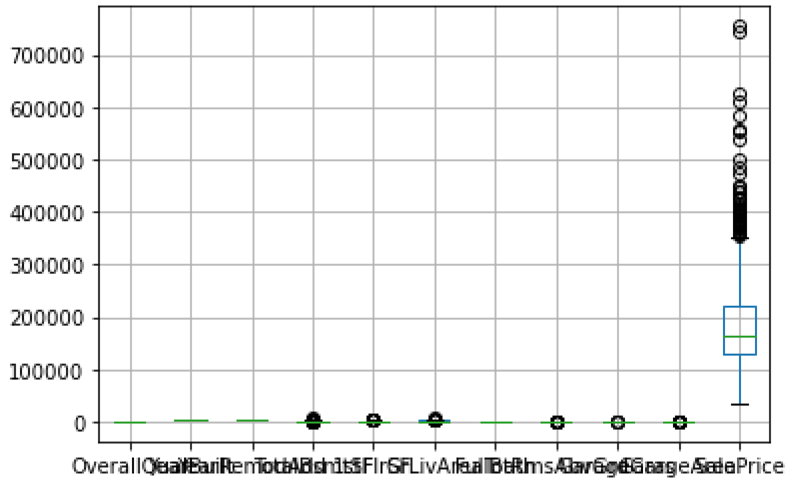
	OverallQual	YearBuilt	YearRemodAdd	TotalBsmtSF	1stFlrSF	GrLivArea	FullBath	TotRm
0	7	2003	2003	856	856	1710	2	
1	6	1976	1976	1262	1262	1262	2	
2	7	2001	2002	920	920	1786	2	
3	7	1915	1970	756	961	1717	1	
4	8	2000	2000	1145	1145	2198	2	

In [27]:

```
relevant_feaures_df.boxplot()
```

Out[27]:

<AxesSubplot:>



In [ ]:

```

```

EDA of Categorical Variables

In [32]:

```
categoric_df = df.select_dtypes(exclude=[np.number])
categorycol = categoric_df.columns.to_list()

print("Category :",categorycol)
```

```
Category : ['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Uti
lities', 'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Conditio
n2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Ex
terior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQua
l', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Heating',
'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functiol', 'Firep
laceQu', 'GarageType', 'GarageFinish', 'GarageQual', 'GarageCond', 'PavedD
rive', 'PoolQC', 'Fence', 'MiscFeature', 'SaleType', 'SaleCondition']
```

In [33]:

```
categoric_df_r =categoric_df[['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour',
categoric_df_r
```

Out[33]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neig
0	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1	RL	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	
2	RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	
3	RL	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	
4	RL	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	
...	...	...	...	...	...	...	...	...	
1455	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1456	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1457	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1458	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1459	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	

1460 rows × 43 columns

In [34]:

```
#TREATING MISSING VALUES OF CATEGOGICAL VARIABLES
categoric_df_r.isna().sum(axis=0)
```

Out[34]:

```
MSZoning            0
Street              0
Alley            1369
LotShape            0
LandContour         0
Utilities           0
LotConfig           0
LandSlope           0
Neighborhood        0
Condition1          0
Condition2          0
BldgType            0
HouseStyle          0
RoofStyle           0
RoofMatl            0
Exterior1st         0
Exterior2nd         0
MasVnrType          8
ExterQual           0
ExterCond           0
Foundation          0
BsmtQual           37
BsmtCond           37
BsmtExposure       38
BsmtFinType1       37
BsmtFinType2       38
Heating             0
HeatingQC           0
CentralAir          0
Electrical          1
KitchenQual         0
Functiol            0
FireplaceQu       690
GarageType         81
GarageFinish       81
GarageQual         81
GarageCond         81
PavedDrive          0
PoolQC           1453
Fence            1179
MiscFeature      1406
SaleType            0
SaleCondition       0
dtype: int64
```

In [ ]:

```

```

In [35]:

```
categoric_df_rc=categoric_df_r.dropna(axis='columns')
categoric_df_rc
```

Out[35]:

	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborho
0	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Collg
1	RL	Pave	Reg	Lvl	AllPub	FR2	Gtl	Veenl
2	RL	Pave	IR1	Lvl	AllPub	Inside	Gtl	Collg
3	RL	Pave	IR1	Lvl	AllPub	Corner	Gtl	Craw
4	RL	Pave	IR1	Lvl	AllPub	FR2	Gtl	NoRic
...	...	...	...	...	...	...	...	
1455	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Gilb
1456	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	NWAm
1457	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Craw
1458	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	m
1459	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Edwa

1460 rows × 27 columns

In [36]:

```
categoric_df_rc.isna().sum(axis=0)
```

Out[36]:

```
MSZoning         0
Street           0
LotShape         0
LandContour      0
Utilities        0
LotConfig        0
LandSlope        0
Neighborhood     0
Condition1       0
Condition2       0
BldgType         0
HouseStyle       0
RoofStyle        0
RoofMatl         0
Exterior1st      0
Exterior2nd      0
ExterQual        0
ExterCond        0
Foundation       0
Heating          0
HeatingQC        0
CentralAir       0
KitchenQual      0
Functiol         0
PavedDrive       0
SaleType         0
SaleCondition    0
dtype: int64
```

In [ ]:

```
#categoric_df_rc.boxplot()
```

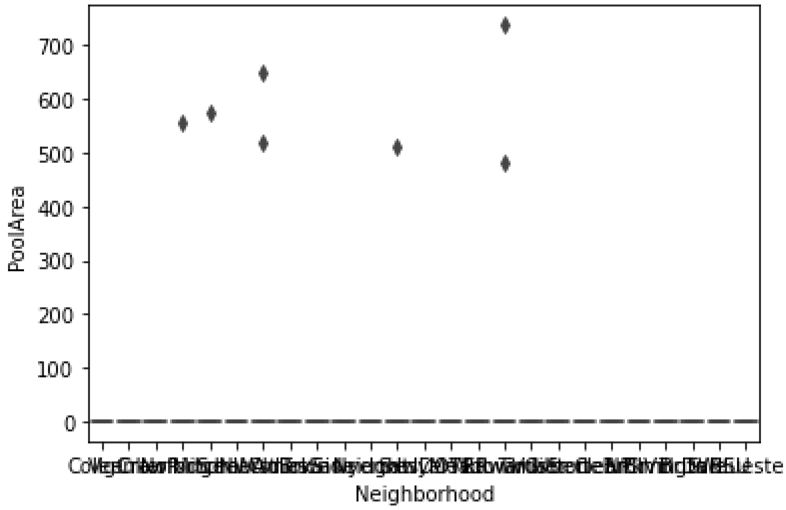
In [37]:

```
categoric_df_rcol = categoric_df_rc.columns.to_list()

print("Category :",categoric_df_rcol)
```

```
Category : ['MSZoning', 'Street', 'LotShape', 'LandContour', 'Utilities',
'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'Bld
gType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2n
d', 'ExterQual', 'ExterCond', 'Foundation', 'Heating', 'HeatingQC', 'Centr
alAir', 'KitchenQual', 'Functiol', 'PavedDrive', 'SaleType', 'SaleConditio
n']
```

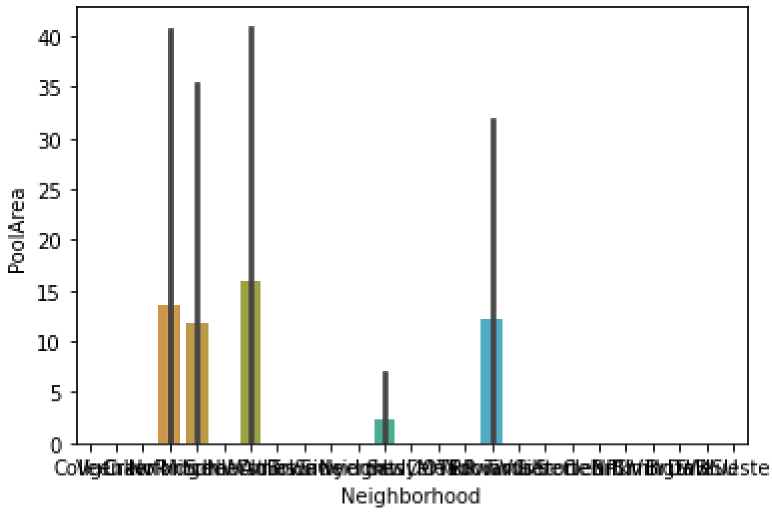
In [38]:

```
#BOXPLOT AND COUNT PLOT OF CATEGORICALVARIALES
import seaborn as sns
#boxplot of categorical varibales
sns.boxplot(x='Neighborhood',y='PoolArea', data=df )
plt.show()
```



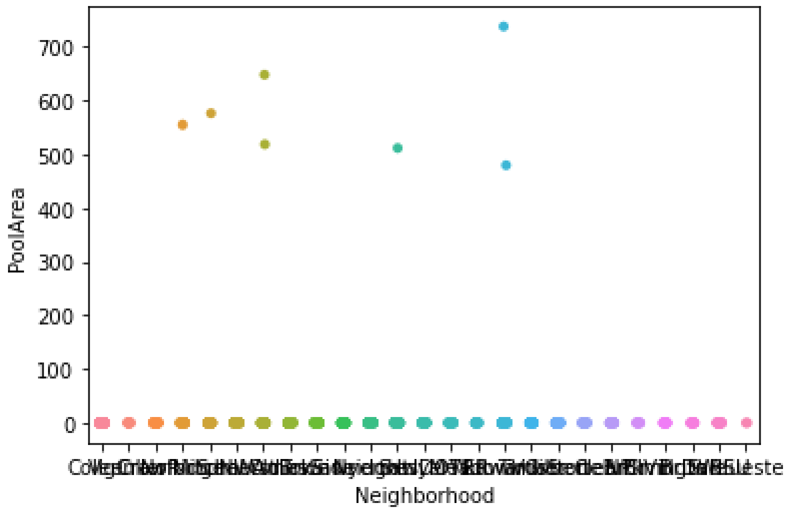
In [39]:

```
sns.barplot(x='Neighborhood',y='PoolArea', data=df )
plt.show()
```



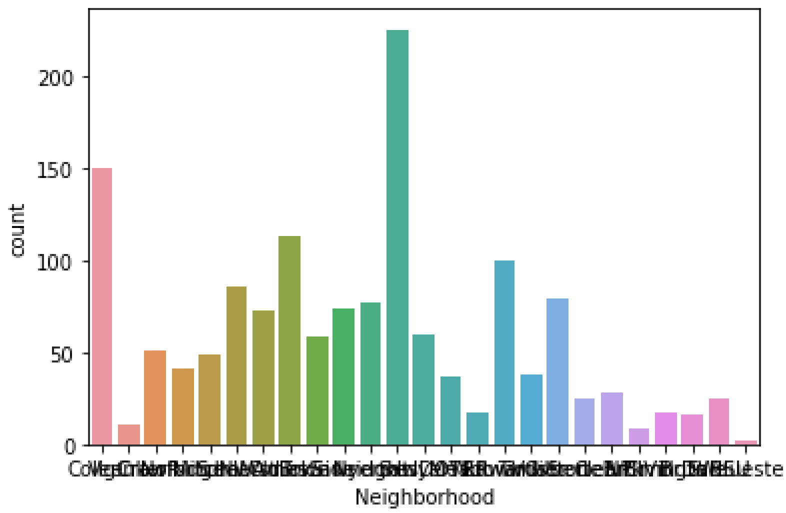
In [40]:

```
sns.stripplot(x='Neighborhood',y='PoolArea', data=df )
plt.show()
```



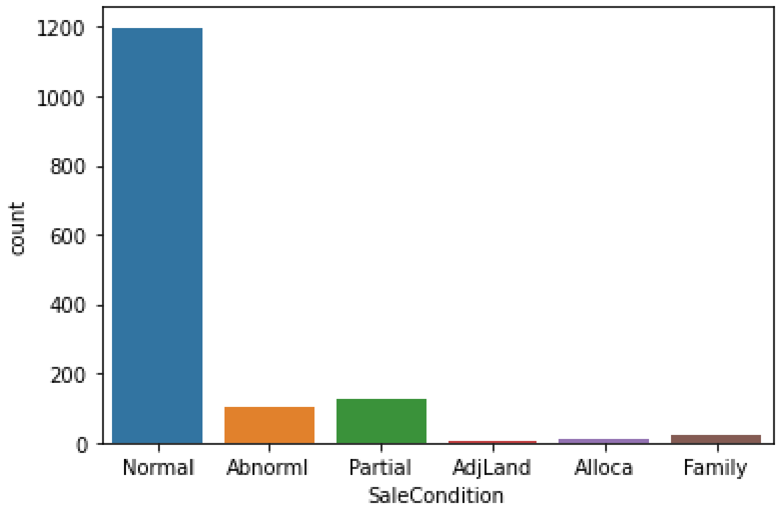
In [41]:

```
#countPlot
import seaborn as sns
import matplotlib.pyplot as plt


sns.countplot(x='Neighborhood', data=df)
plt.show()
```



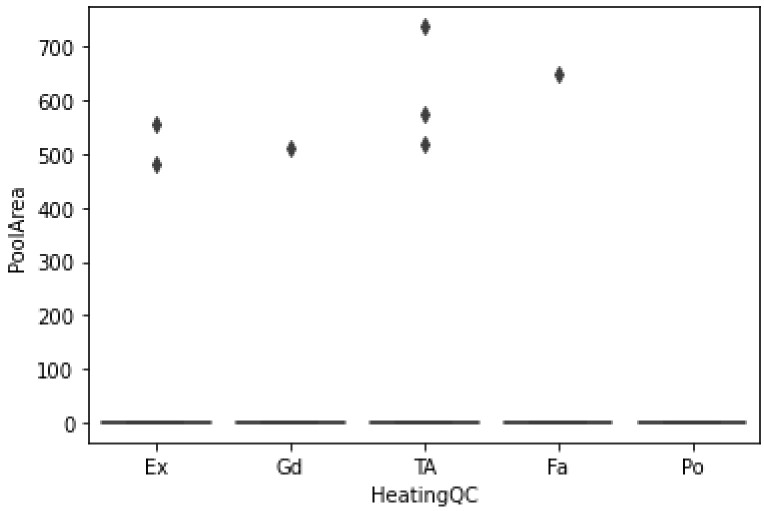
In [42]:

```
#countPlot
import seaborn as sns
import matplotlib.pyplot as plt


sns.countplot(x='SaleCondition', data=df)
plt.show()
```



In [43]:

```
import seaborn as sns
#boxplot of categorical varibales
sns.boxplot(x='HeatingQC',y='PoolArea', data=df )
plt.show()
```



In [44]:

```
#chi-square for relevant features of categorical variables
```

In [45]:

```
categoric_df_rc
```

Out[45]:

	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborho
0	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Collg
1	RL	Pave	Reg	Lvl	AllPub	FR2	Gtl	Veenl
2	RL	Pave	IR1	Lvl	AllPub	Inside	Gtl	Collg
3	RL	Pave	IR1	Lvl	AllPub	Corner	Gtl	Craw
4	RL	Pave	IR1	Lvl	AllPub	FR2	Gtl	NoRic
...	...	...	...	...	...	...	...	
1455	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Gilb
1456	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	NWAr
1457	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Craw
1458	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	r
1459	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	Edwa

1460 rows × 27 columns

In [46]:

```
#fill null values
for col in categoric_df_rc.columns:
    categoric_df_rc[col] =categoric_df_rc[col].fillna(categoric_df_rc[col].mode()[0])
categoric_df_rc.head()
```

Out[46]:

	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood
0	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	CollgCr
1	RL	Pave	Reg	Lvl	AllPub	FR2	Gtl	Veenker
2	RL	Pave	IR1	Lvl	AllPub	Inside	Gtl	CollgCr
3	RL	Pave	IR1	Lvl	AllPub	Corner	Gtl	Crawfor
4	RL	Pave	IR1	Lvl	AllPub	FR2	Gtl	NoRidge

5 rows × 27 columns

In [ ]:

In [47]:

```
#Label Encoder
from sklearn.preprocessing import LabelEncoder
for col in categoric_df_rc.columns:
    le = LabelEncoder()
    categoric_df_rc[col] = le.fit_transform(df[col])
categoric_df_rc.head()
```

Out[47]:

	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood
0	3	1	3	3	0	4	0	5
1	3	1	3	3	0	2	0	23
2	3	1	0	3	0	4	0	5
3	3	1	0	3	0	0	0	6
4	3	1	0	3	0	2	0	14

5 rows × 27 columns

In [48]:

```
from sklearn.feature_selection import chi2
X=categoric_df_rc.drop(columns=['SaleCondition'],axis=1)
y=categoric_df_rc['SaleCondition']
```

In [49]:

```
#CHI VALUES AND P VALUES
chi_scores = chi2(X,y)
chi_scores
```

Out[49]:

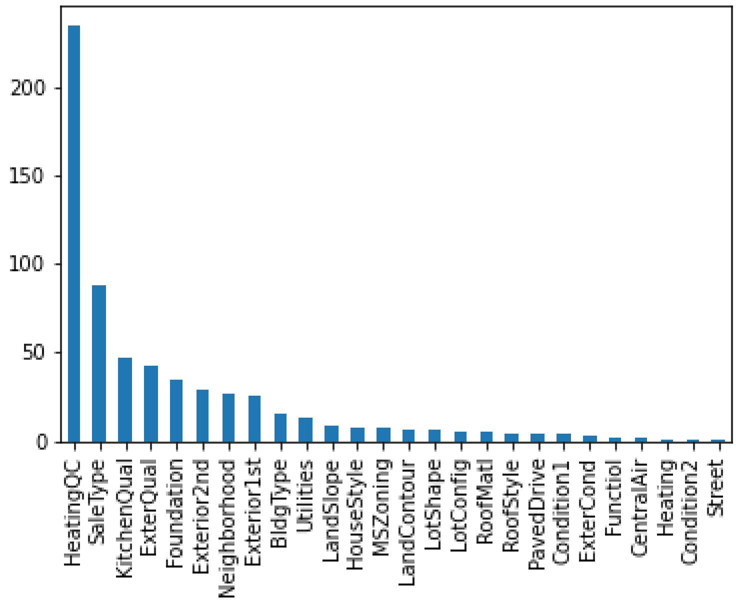
```
(array([6.97882668e+00, 7.98721543e-02, 6.28159657e+00, 6.37729026e+00,
        1.34554455e+01, 5.44281867e+00, 8.92601177e+00, 2.63508220e+01,
        3.50827574e+00, 1.09291080e-01, 1.56917578e+01, 7.40353700e+00,
        4.52856695e+00, 4.61671727e+00, 2.59343763e+01, 2.86197126e+01,
        4.27335186e+01, 3.11669837e+00, 3.49556249e+01, 2.43599707e-01,
        2.34023196e+02, 1.53397607e+00, 4.74130547e+01, 1.67739753e+00,
        4.28159390e+00, 8.80913653e+01]),
 array([2.22219764e-01, 9.99906790e-01, 2.79774578e-01, 2.71217914e-01,
        1.94647933e-02, 3.64256770e-01, 1.12051941e-01, 7.62871948e-05,
        6.22135775e-01, 9.99797983e-01, 7.78150362e-03, 1.92316466e-01,
        4.76070401e-01, 4.64420534e-01, 9.18893575e-05, 2.75302058e-05,
        4.18430801e-08, 6.81999918e-01, 1.53564549e-06, 9.98571421e-01,
        1.46815269e-48, 9.09115996e-01, 4.67983662e-09, 8.91734892e-01,
        5.09625675e-01, 1.69093266e-17]))
```

In [50]:

```
#the higher the chi values the higher the importance
chi_values = pd.Series(chi_scores[0], index=X.columns)
chi_values.sort_values(ascending=False, inplace=True)
chi_values.plot.bar()
```

Out[50]:

<AxesSubplot:>

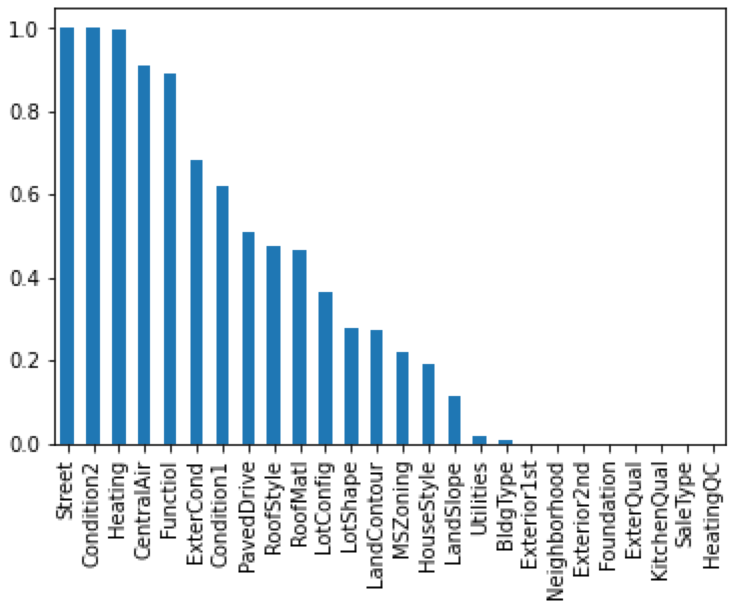


In [51]:

```
#the higher the p values the lower the importance
p_values = pd.Series(chi_scores[1], index=X.columns)
p_values.sort_values(ascending=False, inplace=True)
p_values.plot.bar()
```

Out[51]:

<AxesSubplot:>



In [52]:

```
#significant variables p value < 0.05

relevant_cat = categoric_df_rc[['HeatingQC','SaleType','KitchenQual','ExterQual','Foundat
```

In [53]:

```
#RELEVANT FEATURES OF CATEGORICAL VARIABLES
relevant_cat.head()
```

Out[53]:

	HeatingQC	SaleType	KitchenQual	ExterQual	Foundation	Exterior2nd	Neighborhood	Ex
0	0	8	2	2	2	13	5	
1	0	8	3	3	1	8	23	
2	0	8	2	2	2	13	5	
3	2	8	2	3	0	15	6	
4	0	8	2	2	2	13	14	

In [54]:

```
relevant_cat.shape
```

Out[54]:

```
(1460, 8)
```

In [ ]:

In [57]:

```
#COMBINED RELEVANT NUMERICAL AND CATEGORICAL FEATURES
final_data = pd.concat([relevant_cat, relevant_feaures_df])
final_data
```

Out[57]:

	HeatingQC	SaleType	KitchenQual	ExterQual	Foundation	Exterior2nd	Neighborhood
0	0.0	8.0	2.0	2.0	2.0	13.0	5.0
1	0.0	8.0	3.0	3.0	1.0	8.0	23.0
2	0.0	8.0	2.0	2.0	2.0	13.0	5.0
3	2.0	8.0	2.0	3.0	0.0	15.0	6.0
4	0.0	8.0	2.0	2.0	2.0	13.0	14.0
...	...	...	...	...	...	...	...
1455	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1456	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1457	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1458	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1459	NaN	NaN	NaN	NaN	NaN	NaN	NaN

2581 rows × 19 columns

In [58]:

```
final_data.shape
```

Out[58]:

(2581, 19)

In [ ]:

```
#boxplot OF COMBINED DATA
```

In [59]:

```
final_data.boxplot()
```

Out[59]:

<AxesSubplot:>