Perform Exploratory Data Analysis to show features that contribute to Sale Price (SalePrice). Write a report on word document explaining your results. Use crosstab, groupby, pandas visualization, matplotlib, seaborn and plotly. (Paste your graphs on word document)

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import plotly.graph_objects as go
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
train=pd.read_csv('C:/Users/user/Downloads/train.csv')
train
```

Out[2]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandConto
0	1	60	RL	65.0	8450	Pave	NaN	Reg	L
1	2	20	RL	80.0	9600	Pave	NaN	Reg	L
2	3	60	RL	68.0	11250	Pave	NaN	IR1	L
3	4	70	RL	60.0	9550	Pave	NaN	IR1	L
4	5	60	RL	84.0	14260	Pave	NaN	IR1	L
...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	L
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	L
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	L
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	L
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	L

1460 rows × 81 columns

In [3]:

```
train.SalePrice.agg(['mean','std','max','min','count','median'])
```

Out[3]:

```
mean      180921.195890
std        79442.502883
max       755000.000000
min        34900.000000
count       1460.000000
median    163000.000000
Name: SalePrice, dtype: float64
```

In [4]:

```
train.describe(include='object').T
```

Out[4]:

	count	unique	top	freq
MSZoning	1460	5	RL	1151
Street	1460	2	Pave	1454
Alley	91	2	Grvl	50
LotShape	1460	4	Reg	925
LandContour	1460	4	Lvl	1311
Utilities	1460	2	AllPub	1459
LotConfig	1460	5	Inside	1052
LandSlope	1460	3	Gtl	1382
Neighborhood	1460	25	NAmes	225
Condition1	1460	9	Norm	1260
Condition2	1460	8	Norm	1445
BldgType	1460	5	1Fam	1220
HouseStyle	1460	8	1Story	726
RoofStyle	1460	6	Gable	1141
RoofMatl	1460	8	CompShg	1434
Exterior1st	1460	15	VinylSd	515
Exterior2nd	1460	16	VinylSd	504
MasVnrType	1452	4	None	864
ExterQual	1460	4	TA	906
ExterCond	1460	5	TA	1282
Foundation	1460	6	PConc	647
BsmtQual	1423	4	TA	649
BsmtCond	1423	4	TA	1311
BsmtExposure	1422	4	No	953
BsmtFinType1	1423	6	Unf	430
BsmtFinType2	1422	6	Unf	1256
Heating	1460	6	GasA	1428
HeatingQC	1460	5	Ex	741
CentralAir	1460	2	Y	1365
Electrical	1459	5	SBrkr	1334
KitchenQual	1460	4	TA	735
Functional	1460	7	Typ	1360
FireplaceQu	770	5	Gd	380
GarageType	1379	6	Attchd	870

	count	unique	top	freq
GarageFinish	1379	3	Unf	605
GarageQual	1379	5	TA	1311
GarageCond	1379	5	TA	1326
PavedDrive	1460	3	Y	1340
PoolQC	7	3	Gd	3
Fence	281	4	MnPrv	157
MiscFeature	54	4	Shed	49
SaleType	1460	9	WD	1267
SaleCondition	1460	6	Normal	1198

In [5]:

```
train.describe(include='object').iloc[:,0:6]
```

Out[5]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities
count	1460	1460	91	1460	1460	1460
unique	5	2	2	4	4	2
top	RL	Pave	Grvl	Reg	Lvl	AllPub
freq	1151	1454	50	925	1311	1459

In [6]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             1460 non-null   int64  
 1   MSSubClass     1460 non-null   int64  
 2   MSZoning       1460 non-null   object 
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64  
 5   Street         1460 non-null   object 
 6   Alley          91 non-null     object 
 7   LotShape       1460 non-null   object 
 8   LandContour    1460 non-null   object 
 9   Utilities      1460 non-null   object 
 10  LotConfig      1460 non-null   object 
 11  LandSlope      1460 non-null   object 
 12  Neighborhood   1460 non-null   object 
 13  Condition1     1460 non-null   object 
 14  Condition2     1460 non-null   object 
 15  BldgType       1460 non-null   object 
 16  HouseStyle     1460 non-null   object 
 17  OverallQual    1460 non-null   int64  
 18  OverallCond    1460 non-null   int64  
 19  YearBuilt      1460 non-null   int64  
 20  YearRemodAdd   1460 non-null   int64  
 21  RoofStyle      1460 non-null   object 
 22  RoofMatl       1460 non-null   object 
 23  Exterior1st    1460 non-null   object 
 24  Exterior2nd    1460 non-null   object 
 25  MasVnrType     1452 non-null   object 
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object 
 28  ExterCond      1460 non-null   object 
 29  Foundation     1460 non-null   object 
 30  BsmtQual       1423 non-null   object 
 31  BsmtCond       1423 non-null   object 
 32  BsmtExposure   1422 non-null   object 
 33  BsmtFinType1   1423 non-null   object 
 34  BsmtFinSF1     1460 non-null   int64  
 35  BsmtFinType2   1422 non-null   object 
 36  BsmtFinSF2     1460 non-null   int64  
 37  BsmtUnfSF      1460 non-null   int64  
 38  TotalBsmtSF    1460 non-null   int64  
 39  Heating        1460 non-null   object 
 40  HeatingQC      1460 non-null   object 
 41  CentralAir     1460 non-null   object 
 42  Electrical     1459 non-null   object 
 43  1stFlrSF       1460 non-null   int64  
 44  2ndFlrSF       1460 non-null   int64  
 45  LowQualFinSF   1460 non-null   int64  
 46  GrLivArea      1460 non-null   int64  
 47  BsmtFullBath   1460 non-null   int64  
 48  BsmtHalfBath   1460 non-null   int64  
 49  FullBath       1460 non-null   int64  
 50  HalfBath       1460 non-null   int64  
```

```
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [7]:

```
train.columns
```

Out[7]:

```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAd
d',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBat
h',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageTyp
e',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQua
l',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [8]:

```
train.nunique()
```

Out[8]:

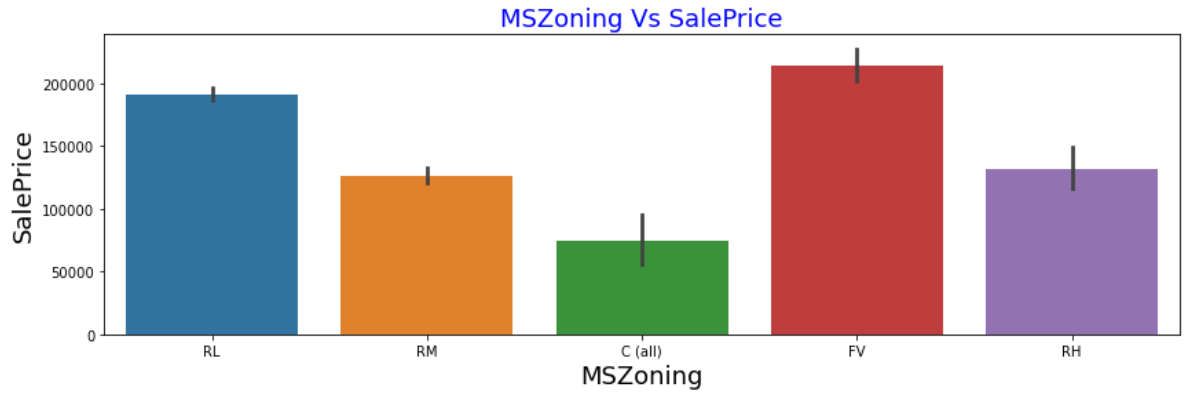
```
Id               1460
MSSubClass         15
MSZoning            5
LotFrontage       110
LotArea          1073
                 ...
MoSold             12
YrSold              5
SaleType            9
SaleCondition       6
SalePrice         663
Length: 81, dtype: int64
```

In [123]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='MSZoning',y='SalePrice',data=train)
plt.title('MSZoning Vs SalePrice',fontsize=18,color='b')
plt.xlabel('MSZoning',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('MSZoning')['SalePrice'].mean()
```

Out[123]:

```
MSZoning
C (all)     74528.000000
FV         214014.061538
RH         131558.375000
RL         191004.994787
RM         126316.830275
Name: SalePrice, dtype: float64
```



In [9]:

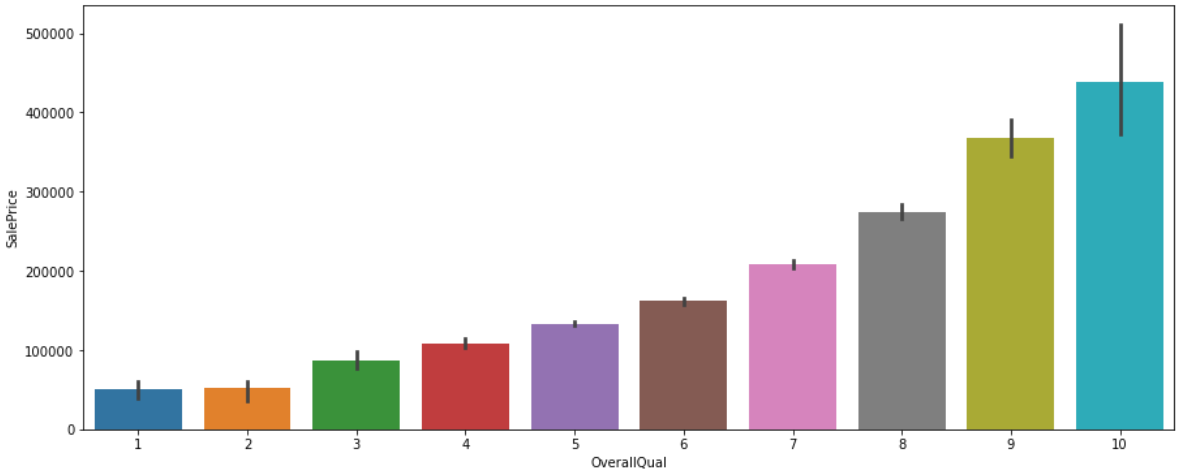
```
trn=train.select_dtypes(include=['float','int'])
trn1=trn.loc[:,trn.nunique()<=10]
trn1.nunique()
```

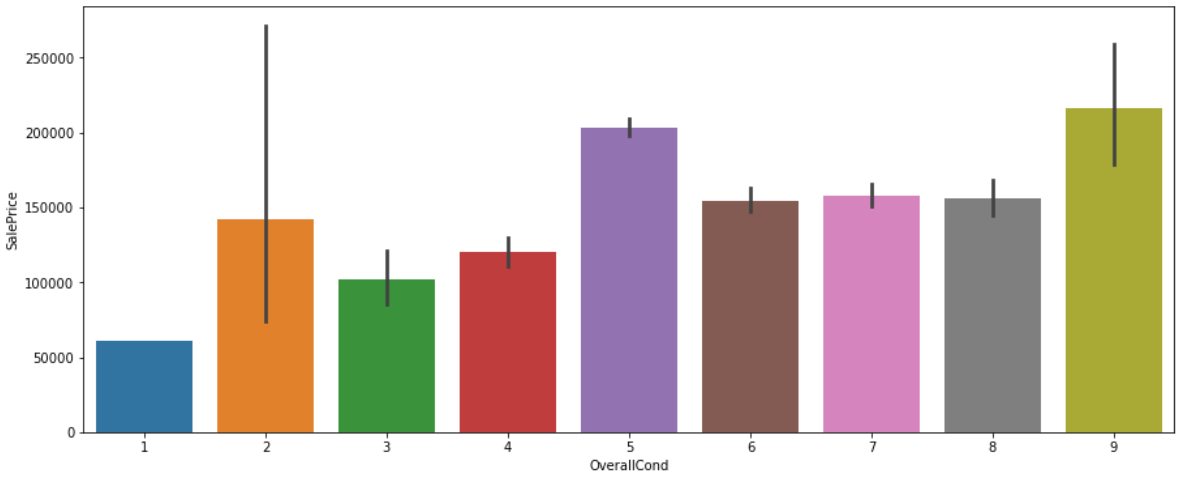
Out[9]:

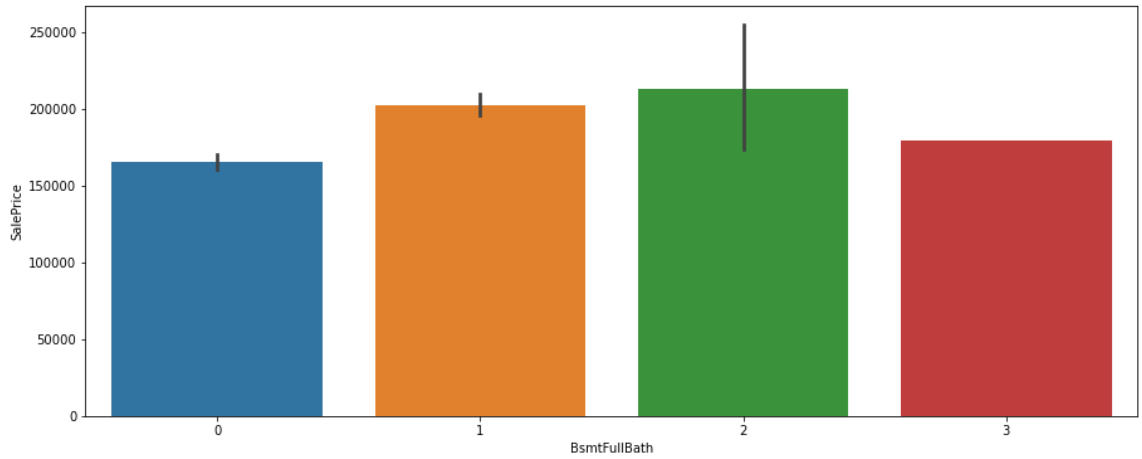
```
OverallQual     10
OverallCond      9
BsmtFullBath     4
BsmtHalfBath     3
FullBath         4
HalfBath         3
BedroomAbvGr     8
KitchenAbvGr     4
Fireplaces       4
GarageCars       5
PoolArea         8
YrSold           5
dtype: int64
```

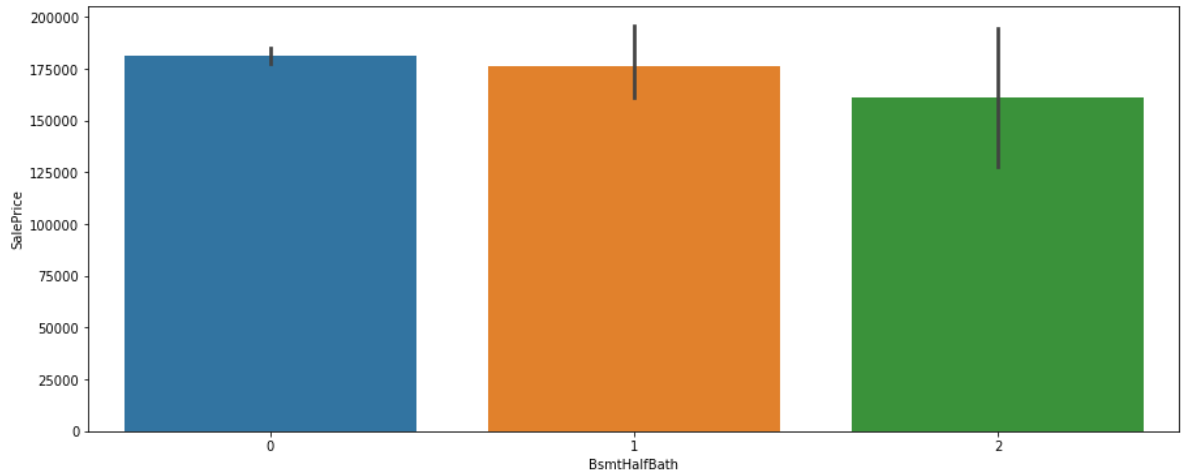
In [10]:

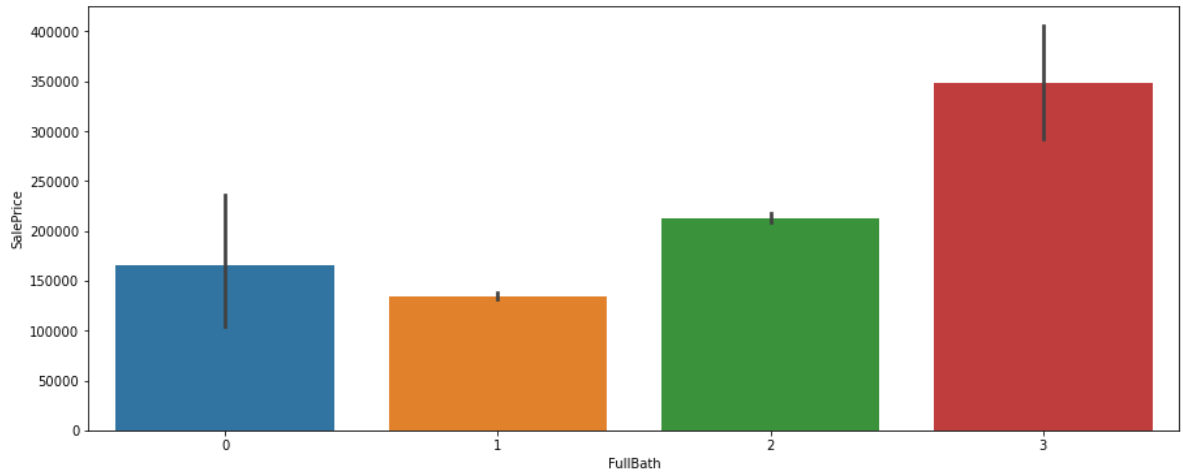
```
for h in trn1.columns:
    plt.figure(figsize=(15,6))
    sb.barplot(x=h,y='SalePrice',data=train);
```

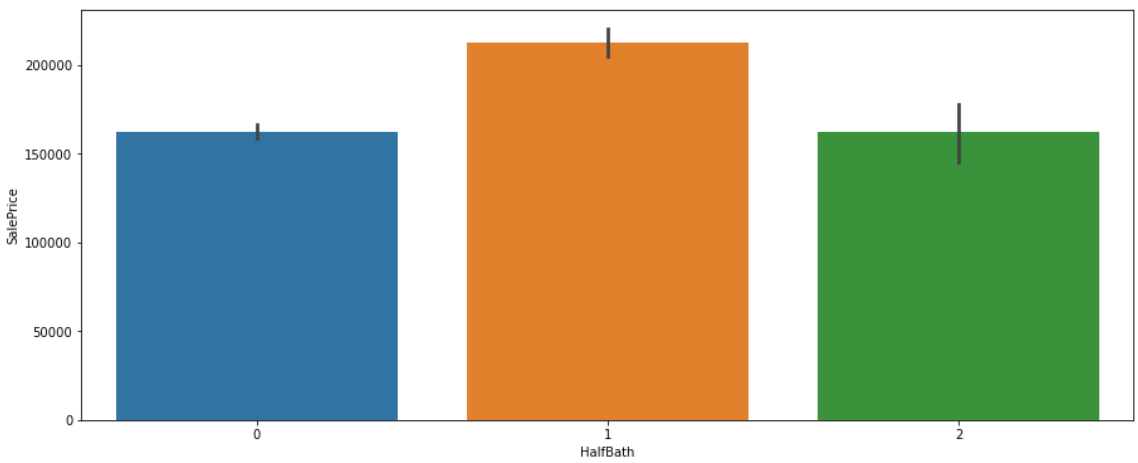


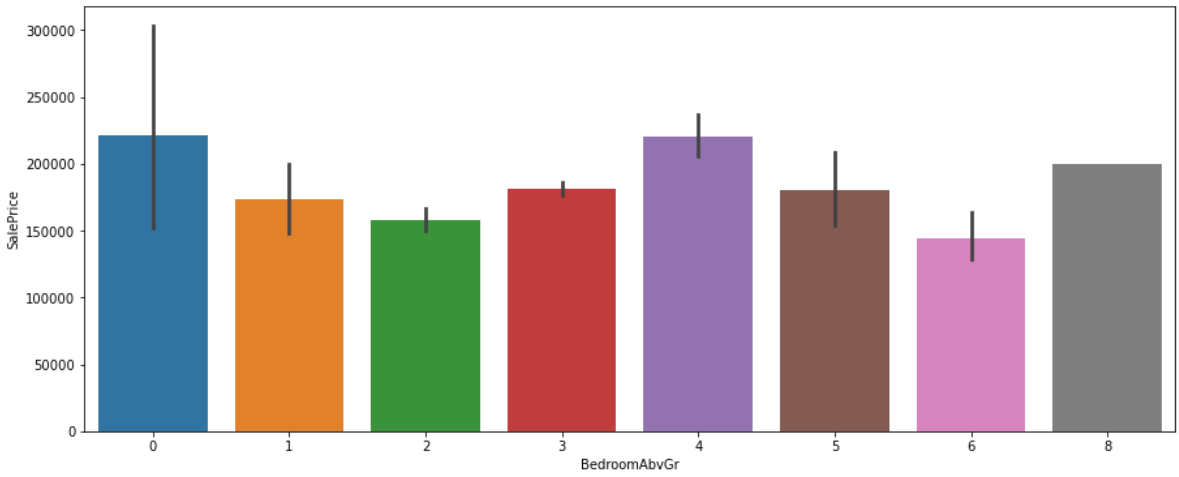


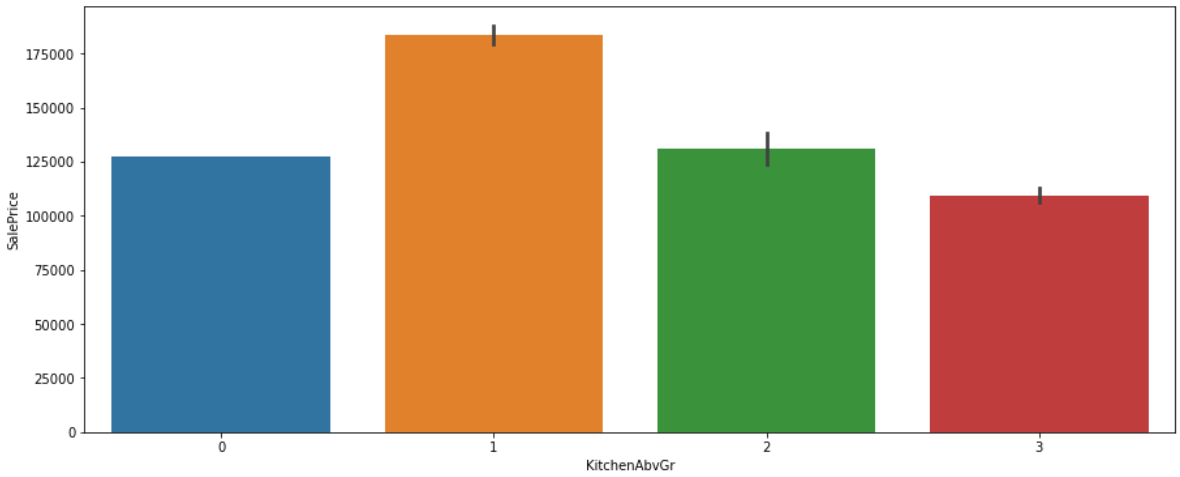

SalePrice
250000
200000
150000
100000
50000
0
0
1
2
3
BsmtFullBath

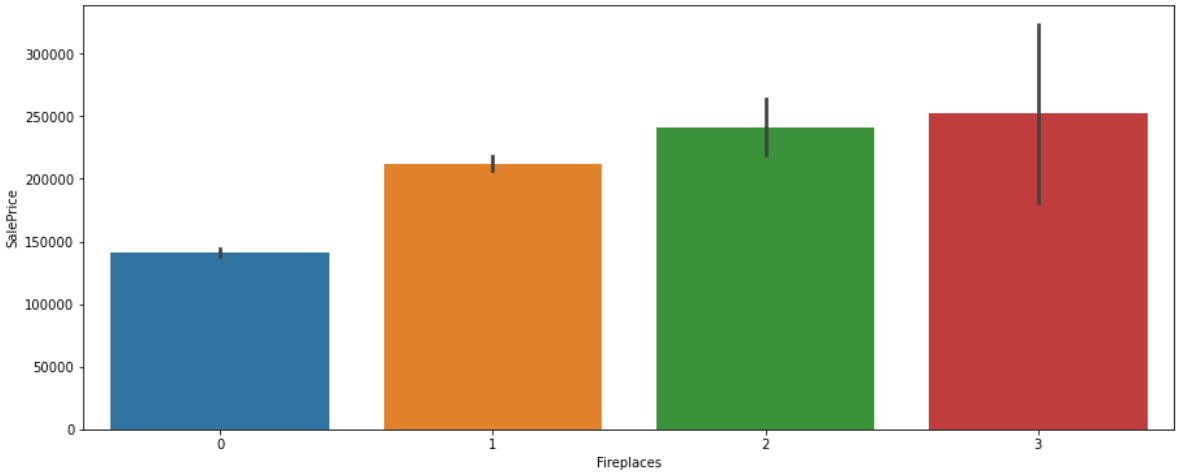

SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
0
1
2
BsmtHalfBath

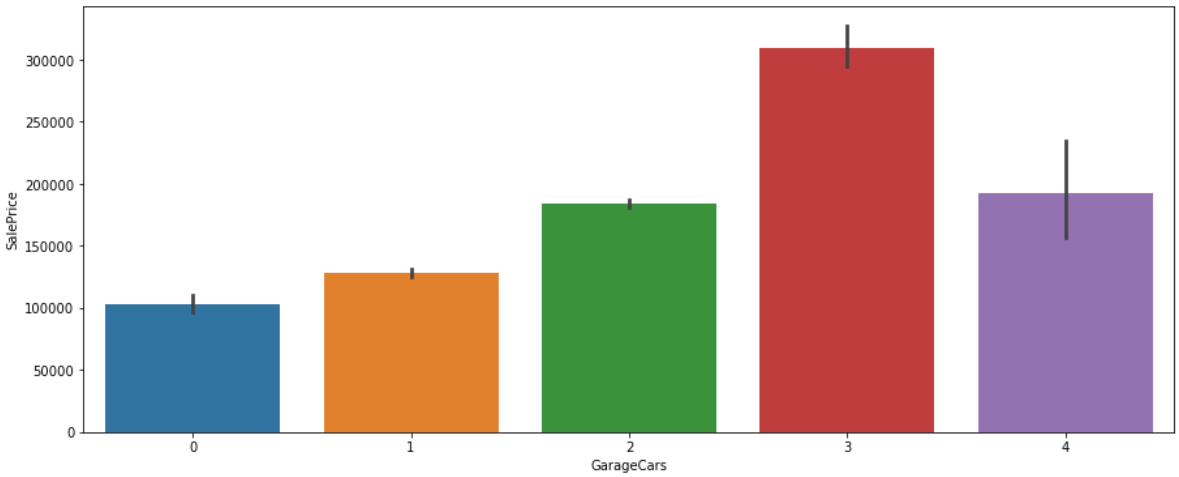

SalePrice
400000
350000
300000
250000
200000
150000
100000
50000
0
0
1
2
3
FullBath

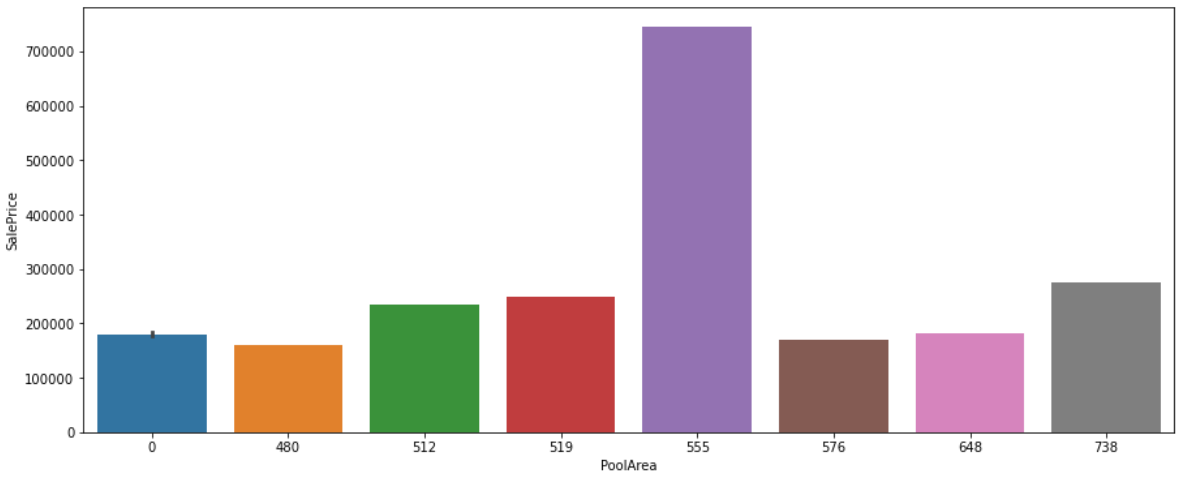

SalePrice
200000
150000
100000
50000
0
0
1
2
HalfBath

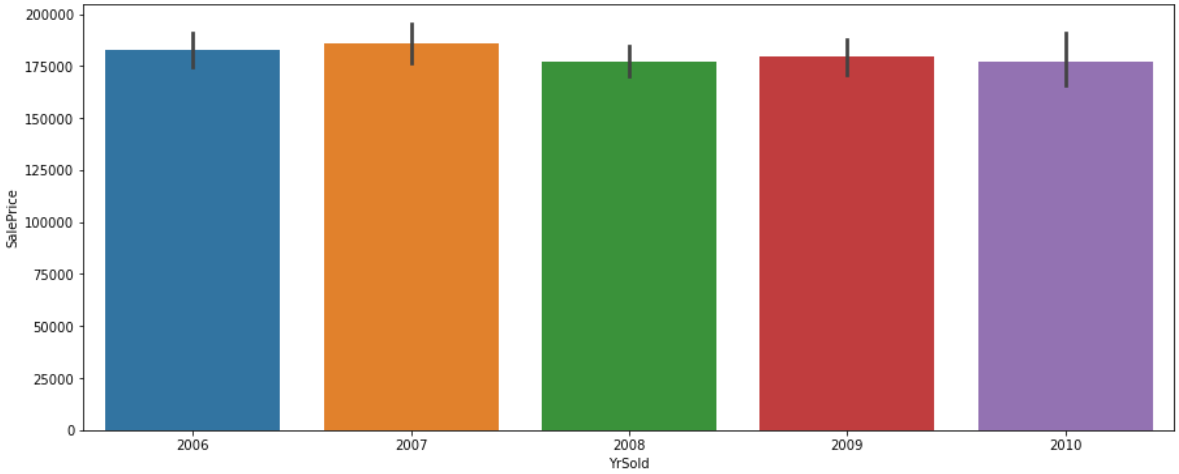

SalePrice
300000
250000
200000
150000
100000
50000
0
0
1
2
3
4
5
6
8
BedroomAbvGr


SalePrice
175000
150000
125000
100000
75000
50000
25000
0
0
1
2
3
KitchenAbvGr


SalePrice
300000
250000
200000
150000
100000
50000
0
0
1
2
3
Fireplaces


SalePrice
300000
250000
200000
150000
100000
50000
0
0
1
2
3
4
GarageCars


SalePrice
700000
600000
500000
400000
300000
200000
100000
0
0
480
512
519
555
576
648
738
PoolArea



In [ ]:

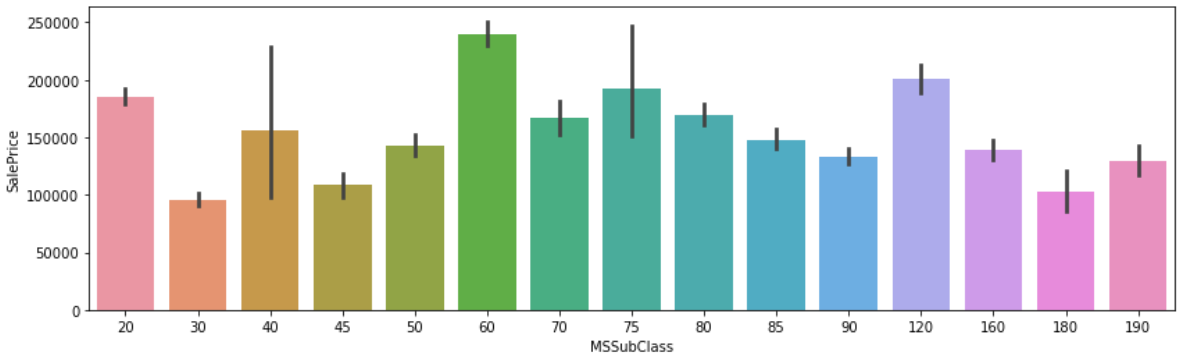
```
```

In [ ]:

```
```

In [11]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='MSSubClass',y='SalePrice',data=train);
```



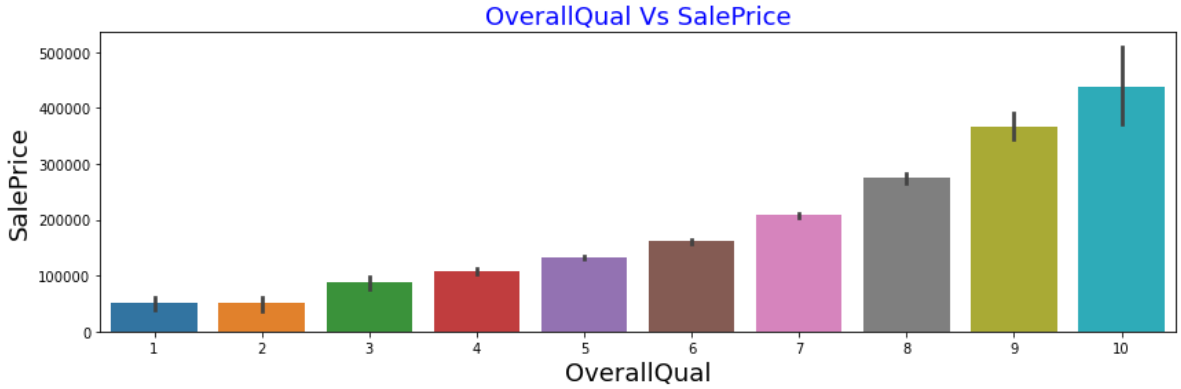
Type *Markdown* and LaTeX: α^2

In [12]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='OverallQual',y='SalePrice',data=train)
plt.title('OverallQual Vs SalePrice',fontsize=18,color='b')
plt.xlabel('OverallQual',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('OverallQual')['SalePrice'].mean()
```

Out[12]:

```
OverallQual
1      50150.000000
2      51770.333333
3      87473.750000
4     108420.655172
5     133523.347607
6     161603.034759
7     207716.423197
8     274735.535714
9     367513.023256
10    438588.388889
Name: SalePrice, dtype: float64
```

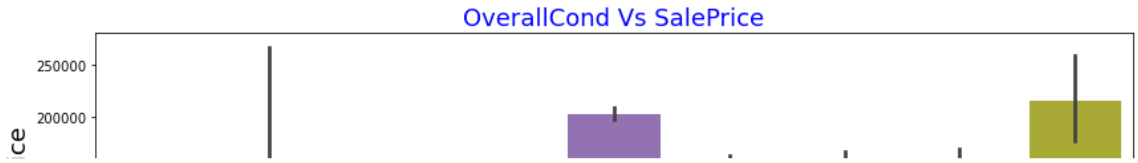


In [13]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='OverallCond',y='SalePrice',data=train)
plt.title('OverallCond Vs SalePrice',fontsize=18,color='b')
plt.xlabel('OverallCond',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('OverallCond')['SalePrice'].mean()
```

Out[13]:

```
OverallCond
1     61000.000000
2    141986.400000
3    101929.400000
4    120438.438596
5    203146.914738
6    153961.591270
7    158145.487805
8    155651.736111
9    216004.545455
Name: SalePrice, dtype: float64
```

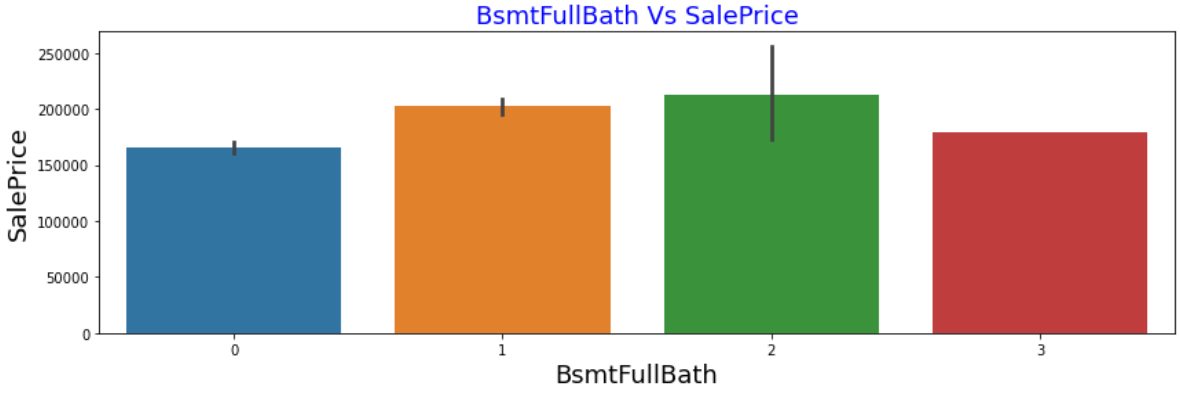


In [14]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='BsmtFullBath',y='SalePrice',data=train);
plt.title('BsmtFullBath Vs SalePrice',fontsize=18,color='b')
plt.xlabel('BsmtFullBath',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('BsmtFullBath')['SalePrice'].mean()
```

Out[14]:

```
BsmtFullBath
0    165521.640187
1    202522.918367
2    213063.066667
3    179000.000000
Name: SalePrice, dtype: float64
```

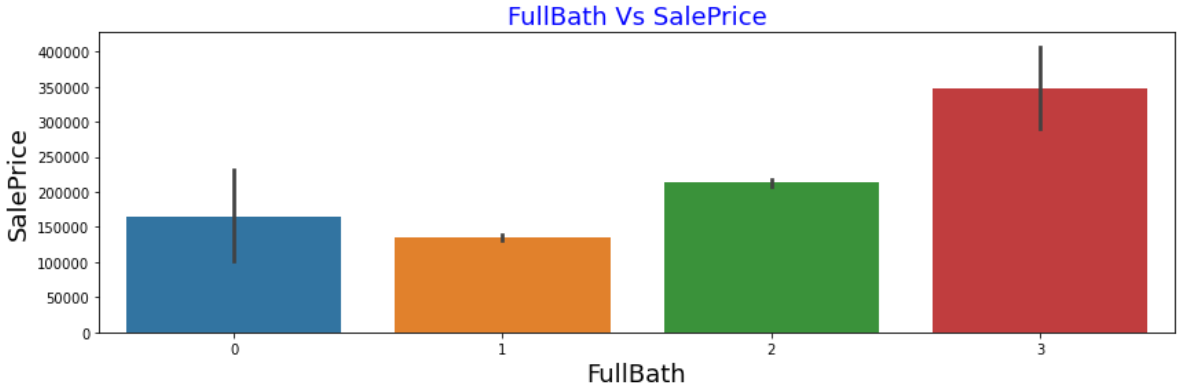


In [15]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='FullBath',y='SalePrice',data=train);
plt.title('FullBath Vs SalePrice',fontsize=18,color='b')
plt.xlabel('FullBath',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('FullBath')['SalePrice'].mean()
```

Out[15]:

```
FullBath
0    165200.888889
1    134751.440000
2    213009.825521
3    347822.909091
Name: SalePrice, dtype: float64
```

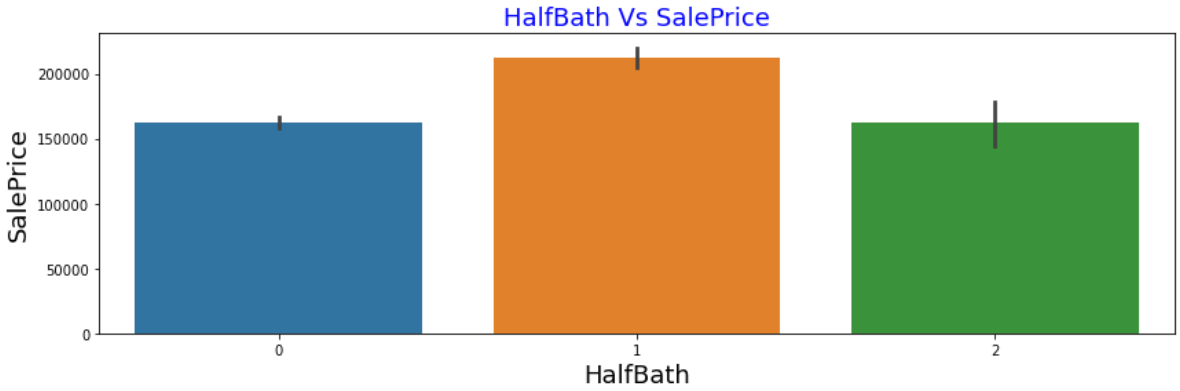


In [49]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='HalfBath',y='SalePrice',data=train);
plt.title('HalfBath Vs SalePrice',fontsize=18,color='b')
plt.xlabel('HalfBath',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('HalfBath')['SalePrice'].mean()
```

Out[49]:

```
HalfBath
0    162534.884995
1    212721.960748
2    162028.916667
Name: SalePrice, dtype: float64
```



In [53]:

```
train.HalfBath.unique()
```

Out[53]:

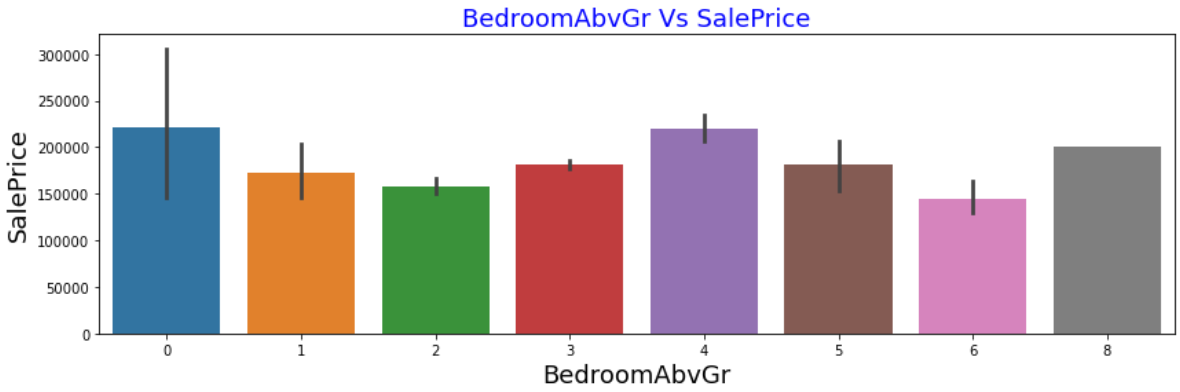
```
array([1, 0, 2], dtype=int64)
```

In [54]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='BedroomAbvGr',y='SalePrice',data=train);
plt.title('BedroomAbvGr Vs SalePrice',fontsize=18,color='b')
plt.xlabel('BedroomAbvGr',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('BedroomAbvGr')['SalePrice'].mean()
```

Out[54]:

```
BedroomAbvGr
0    221493.166667
1    173162.420000
2    158197.659218
3    181056.870647
4    220421.253521
5    180819.047619
6    143779.000000
8    200000.000000
Name: SalePrice, dtype: float64
```

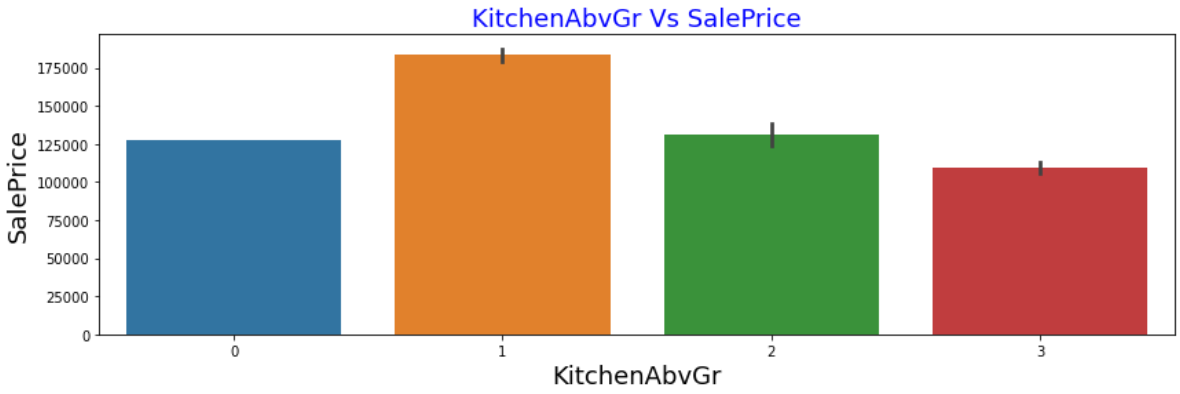


In [55]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='KitchenAbvGr',y='SalePrice',data=train);
plt.title('KitchenAbvGr Vs SalePrice',fontsize=18,color='b')
plt.xlabel('KitchenAbvGr',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('KitchenAbvGr')['SalePrice'].mean()
```

Out[55]:

```
KitchenAbvGr
0    127500.000000
1    183388.790230
2    131096.153846
3    109500.000000
Name: SalePrice, dtype: float64
```

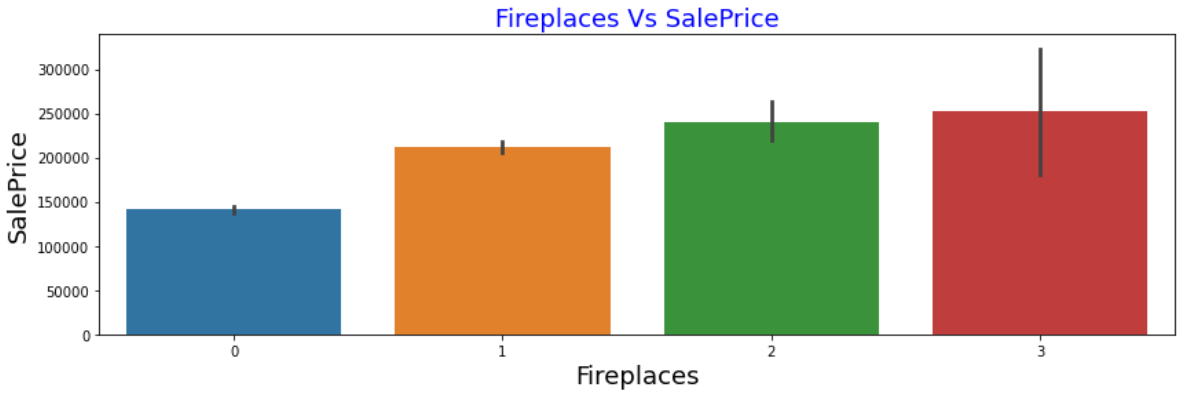


In [56]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='Fireplaces',y='SalePrice',data=train);
plt.title('Fireplaces Vs SalePrice',fontsize=18,color='b')
plt.xlabel('Fireplaces',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('Fireplaces')['SalePrice'].mean()
```

Out[56]:

```
Fireplaces
0    141331.482609
1    211843.909231
2    240588.539130
3    252000.000000
Name: SalePrice, dtype: float64
```

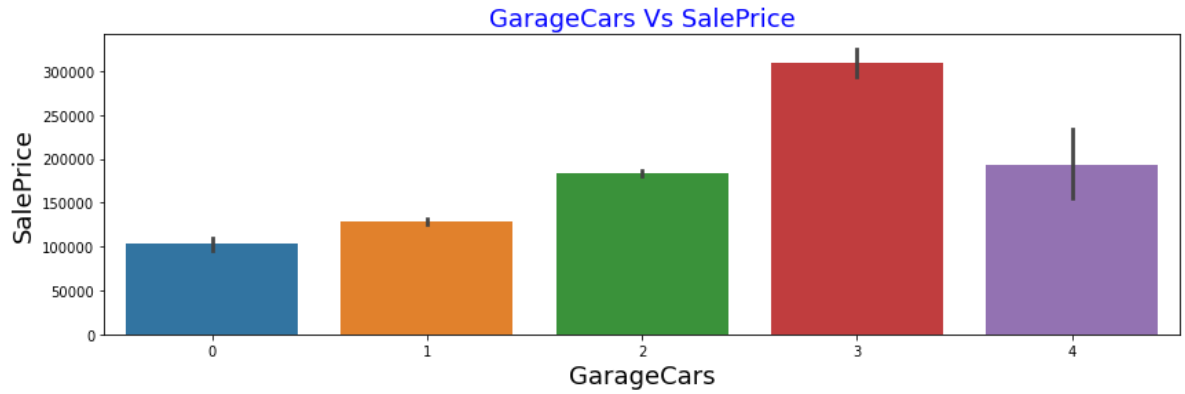


In [57]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='GarageCars',y='SalePrice',data=train);
plt.title('GarageCars Vs SalePrice',fontsize=18,color='b')
plt.xlabel('GarageCars',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('GarageCars')['SalePrice'].mean()
```

Out[57]:

```
GarageCars
0    103317.283951
1    128116.688347
2    183851.663835
3    309636.121547
4    192655.800000
Name: SalePrice, dtype: float64
```

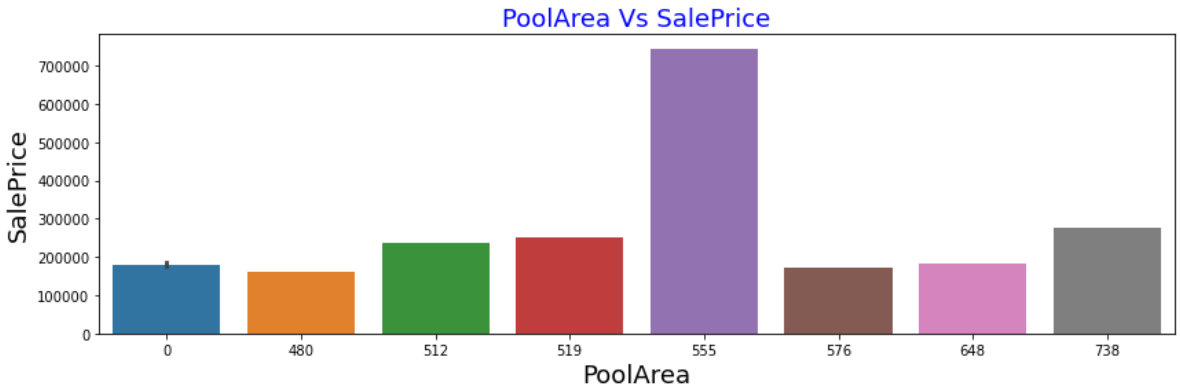


In [59]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='PoolArea',y='SalePrice',data=train);
plt.title('PoolArea Vs SalePrice',fontsize=18,color='b')
plt.xlabel('PoolArea',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('PoolArea')['SalePrice'].mean().round(2)
```

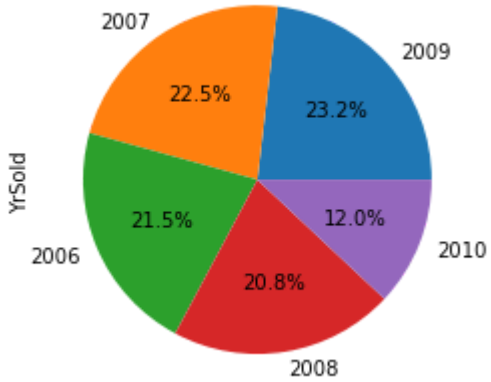
Out[59]:

```
PoolArea
0      180404.66
480    160000.00
512    235000.00
519    250000.00
555    745000.00
576    171000.00
648    181000.00
738    274970.00
Name: SalePrice, dtype: float64
```



In [61]:

```
train.YrSold.value_counts().plot.pie(autopct='%1.1f%%');
```

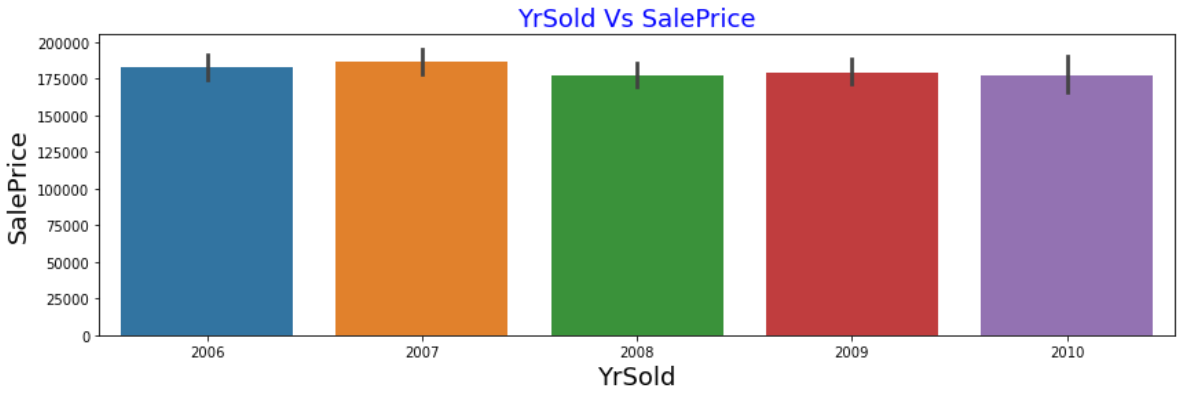


In [62]:

```
plt.figure(figsize=(14,4))
sb.barplot(x='YrSold',y='SalePrice',data=train);
plt.title('YrSold Vs SalePrice',fontsize=18,color='b')
plt.xlabel('YrSold',fontsize=18,color='k')
plt.ylabel('SalePrice',fontsize=18,color='k');
train.groupby('YrSold')['SalePrice'].mean().round(2)
```

Out[62]:

```
YrSold
2006    182549.46
2007    186063.15
2008    177360.84
2009    179432.10
2010    177393.67
Name: SalePrice, dtype: float64
```



In [126]:

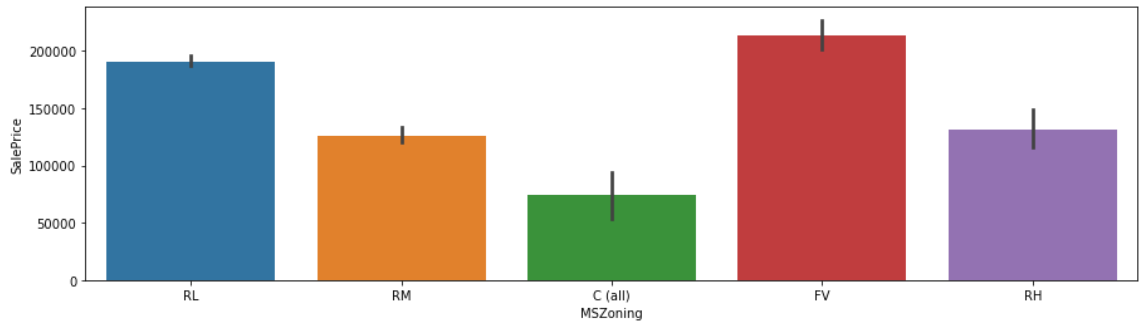
```
t=train.select_dtypes(include='object')
```

Out[126]:

```
MSZoning          5
Street            2
Alley             2
LotShape          4
LandContour       4
Utilities         2
LotConfig         5
LandSlope         3
Neighborhood     25
Condition1        9
Condition2        8
BldgType          5
HouseStyle        8
RoofStyle         6
RoofMatl          8
Exterior1st      15
Exterior2nd      16
MasVnrType        4
ExterQual         4
ExterCond         5
Foundation        6
BsmtQual          4
BsmtCond          4
BsmtExposure      4
BsmtFinType1      6
BsmtFinType2      6
Heating           6
HeatingQC         5
CentralAir        2
Electrical        5
KitchenQual       4
Functional        7
FireplaceQu       5
GarageType        6
GarageFinish      3
GarageQual        5
GarageCond        5
PavedDrive        3
PoolQC            3
Fence             4
MiscFeature       4
SaleType          9
SaleCondition     6
dtype: int64
```

In [127]:

```
for h in t.columns:
    plt.figure(figsize=(15,4))
    sb.barplot(x=h,y='SalePrice',data=train)
```



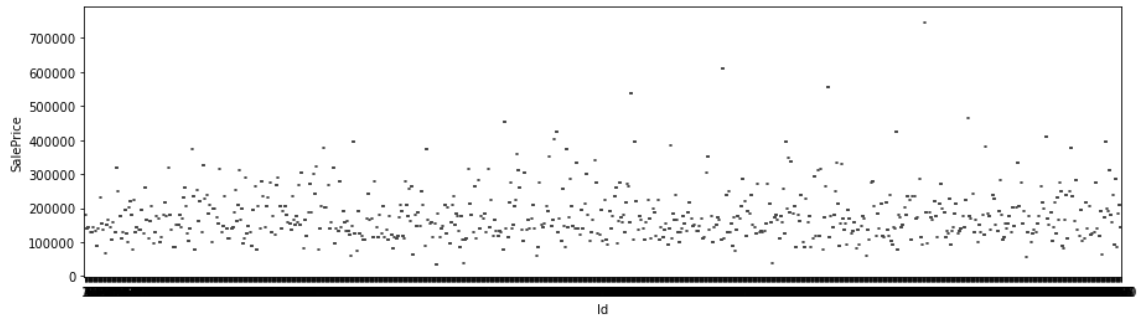
In [24]:

```
trn=train.select_dtypes(include=['float','int'])
trn1=trn.loc[:,trn.nunique()>10]
trn1.nunique()
```

. . .

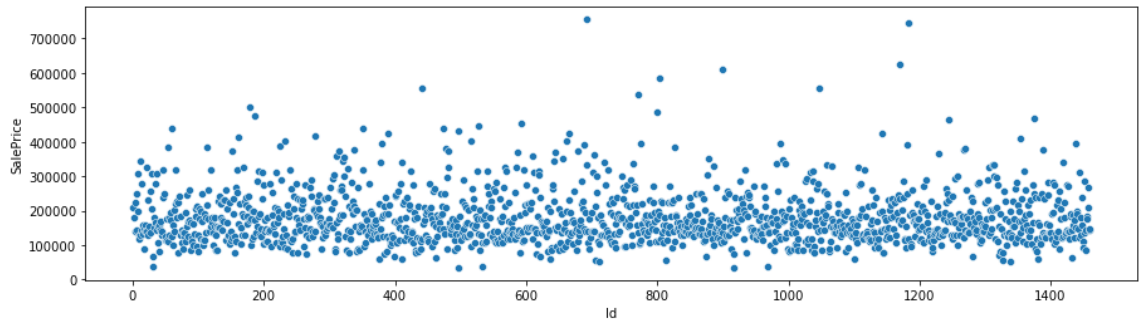
In [25]:

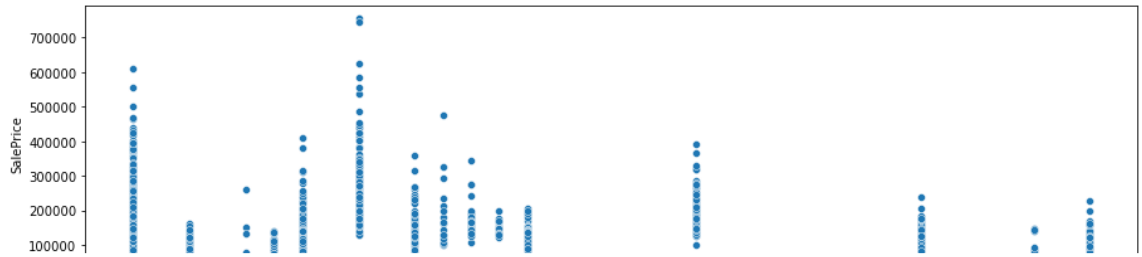
```
for k in trn1.columns:
    plt.figure(figsize=(15,4))
    sb.boxplot(x=k,y='SalePrice',data=train);
```



In [26]:

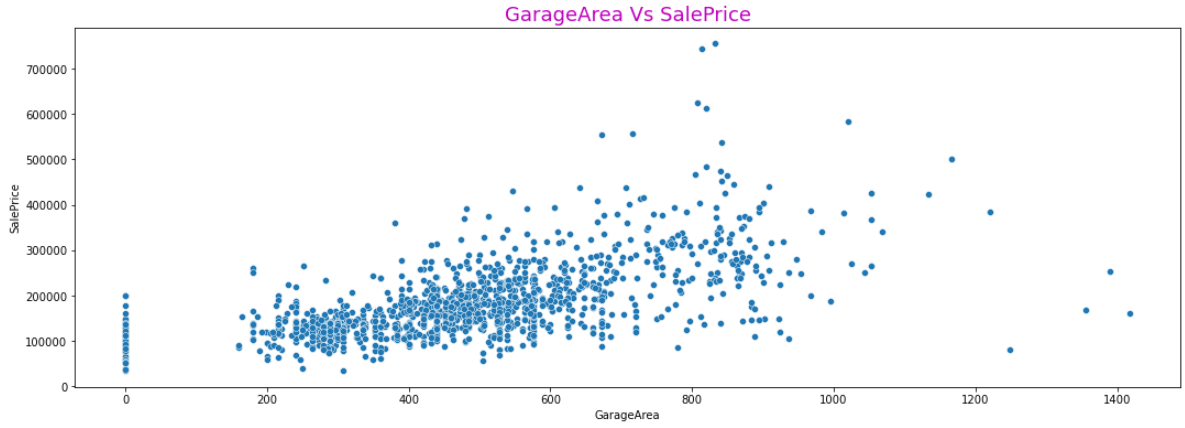
```
for k in trn1.columns:
    plt.figure(figsize=(15,4))
    sb.scatterplot(x=k,y='SalePrice',data=train);
```





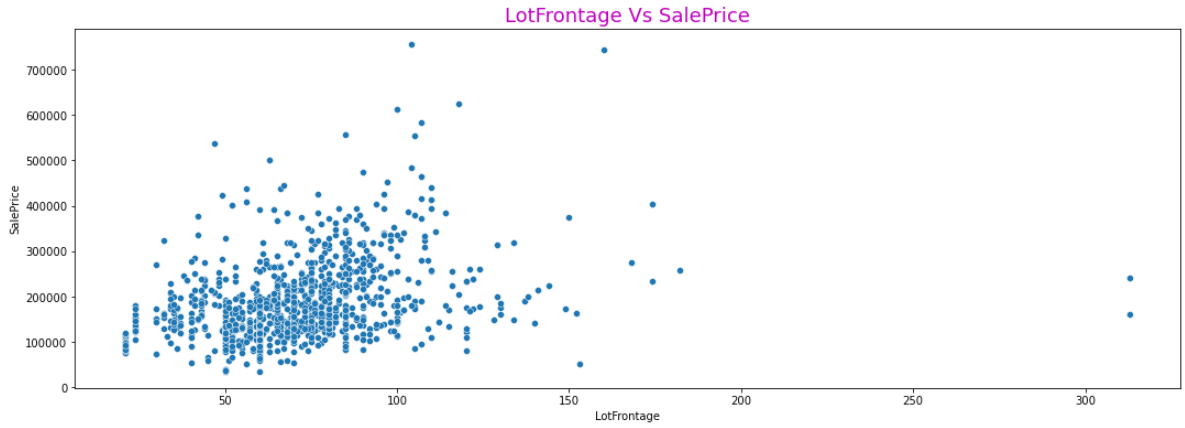
In [102]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="GarageArea",y='SalePrice',data=train);
plt.title('GarageArea Vs SalePrice',fontsize=18,color='m');
```



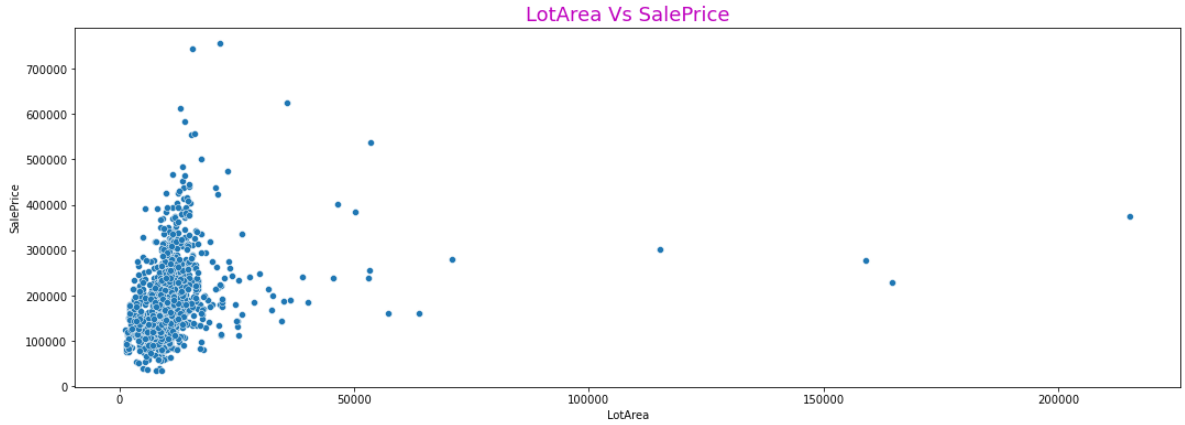
In [101]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="LotFrontage",y='SalePrice',data=train);
plt.title('LotFrontage Vs SalePrice',fontsize=18,color='m');
```



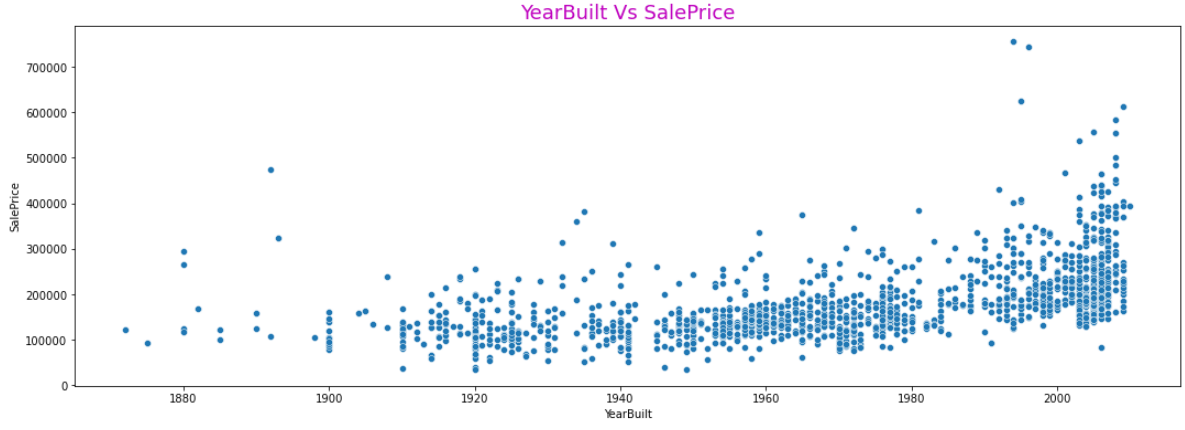
In [103]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="LotArea",y='SalePrice',data=train);
plt.title('LotArea Vs SalePrice',fontsize=18,color='m');
```



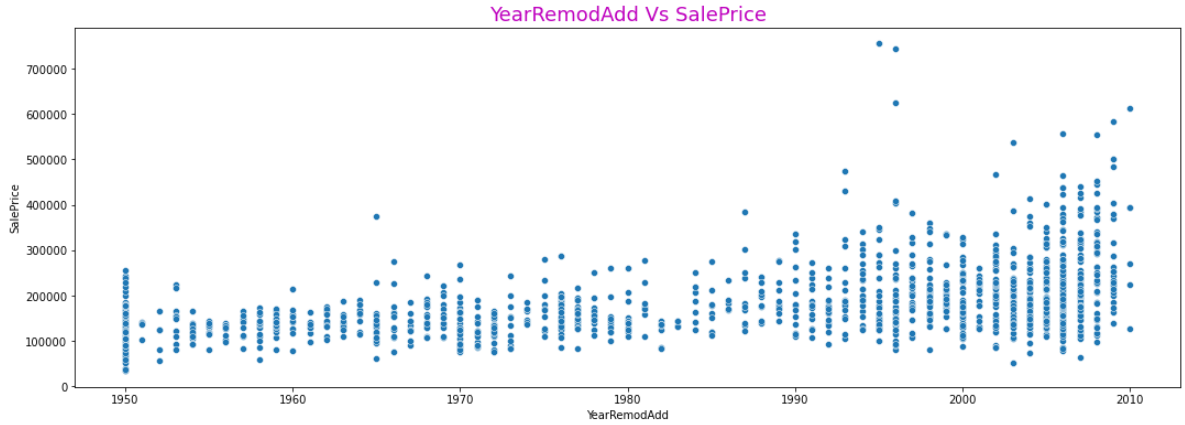
In [104]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="YearBuilt",y='SalePrice',data=train);
plt.title('YearBuilt Vs SalePrice',fontsize=18,color='m');
```



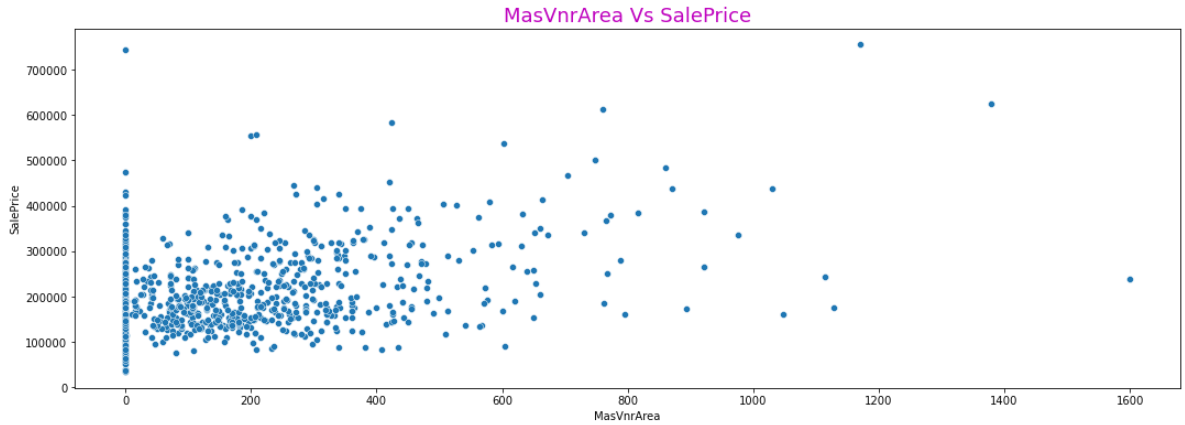
In [105]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="YearRemodAdd",y='SalePrice',data=train);
plt.title('YearRemodAdd Vs SalePrice',fontsize=18,color='m');
```



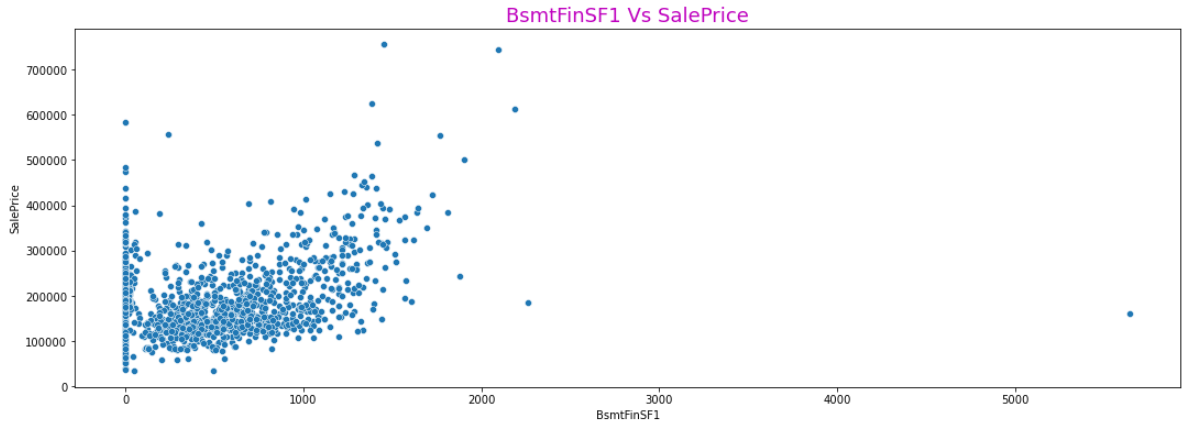
In [106]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="MasVnrArea",y='SalePrice',data=train);
plt.title('MasVnrArea Vs SalePrice',fontsize=18,color='m');
```



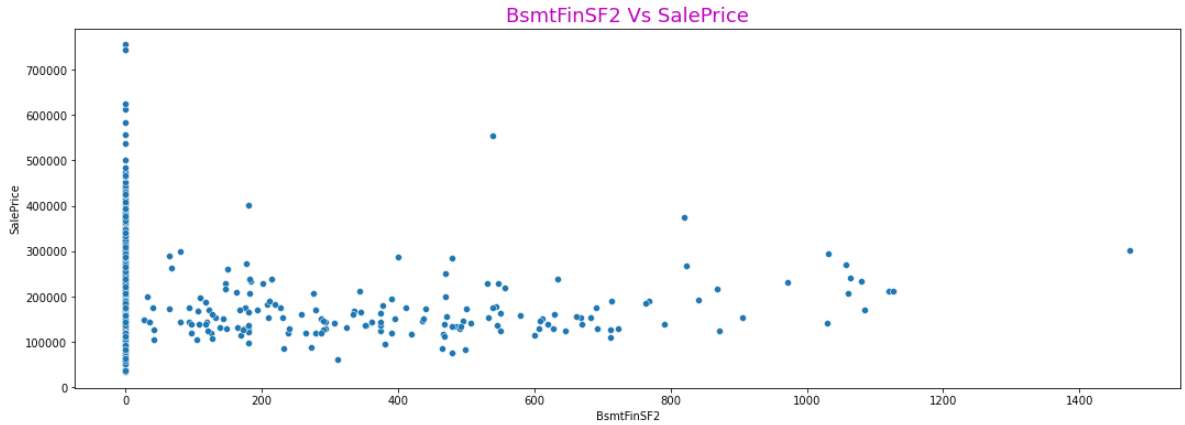
In [107]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="BsmtFinSF1",y='SalePrice',data=train);
plt.title('BsmtFinSF1 Vs SalePrice',fontsize=18,color='m');
```



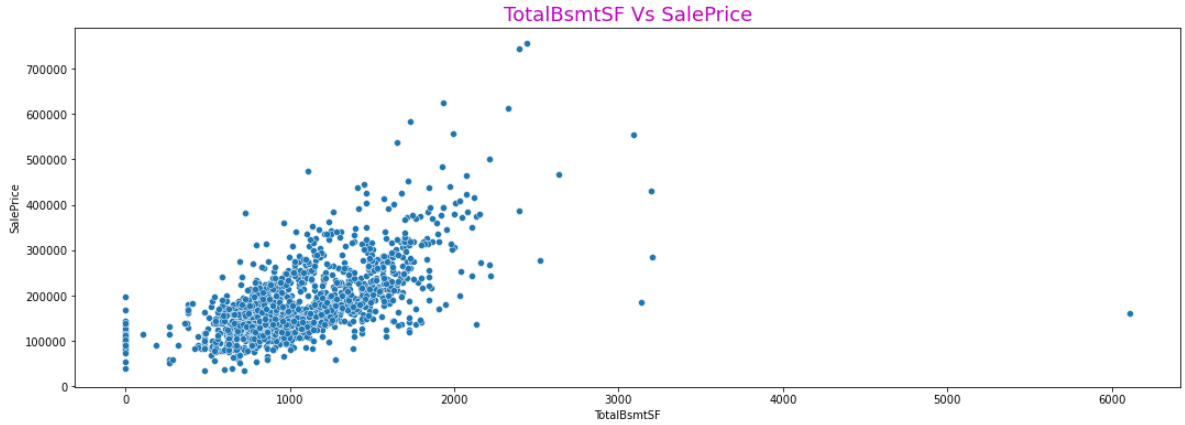
In [108]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="BsmtFinSF2",y='SalePrice',data=train);
plt.title('BsmtFinSF2 Vs SalePrice',fontsize=18,color='m');
```



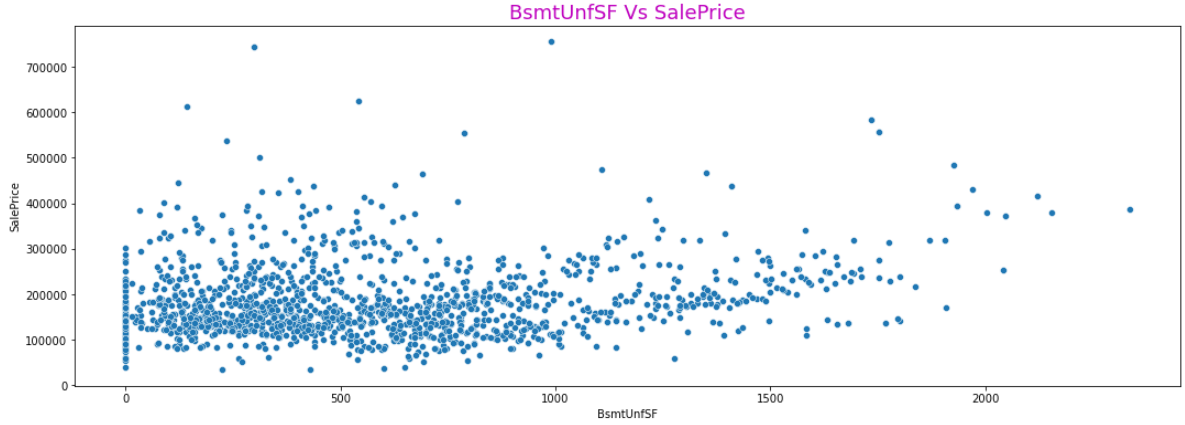
In [109]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="TotalBsmtSF",y='SalePrice',data=train);
plt.title('TotalBsmtSF Vs SalePrice',fontsize=18,color='m');
```



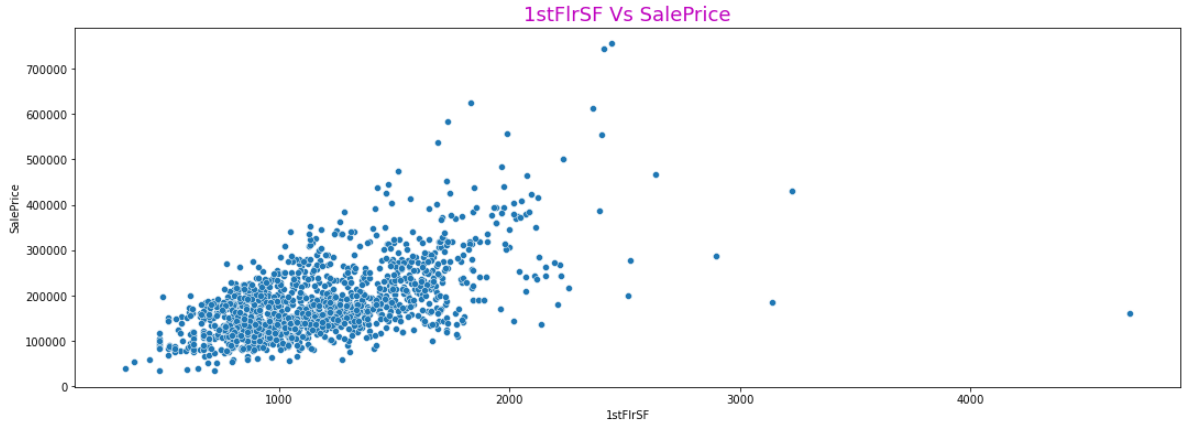
In [110]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="BsmtUnfSF",y='SalePrice',data=train);
plt.title('BsmtUnfSF Vs SalePrice',fontsize=18,color='m');
```



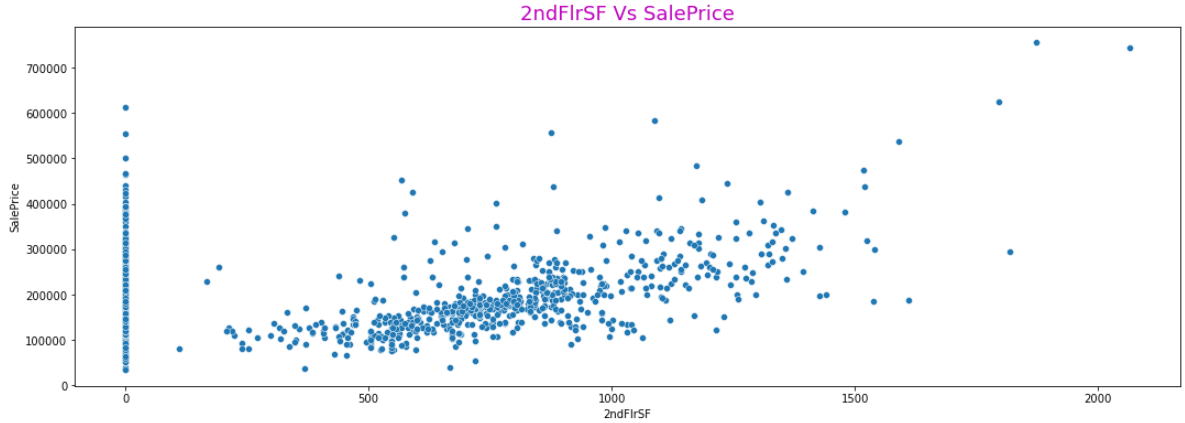
In [111]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="1stFlrSF",y='SalePrice',data=train);
plt.title('1stFlrSF Vs SalePrice',fontsize=18,color='m');
```



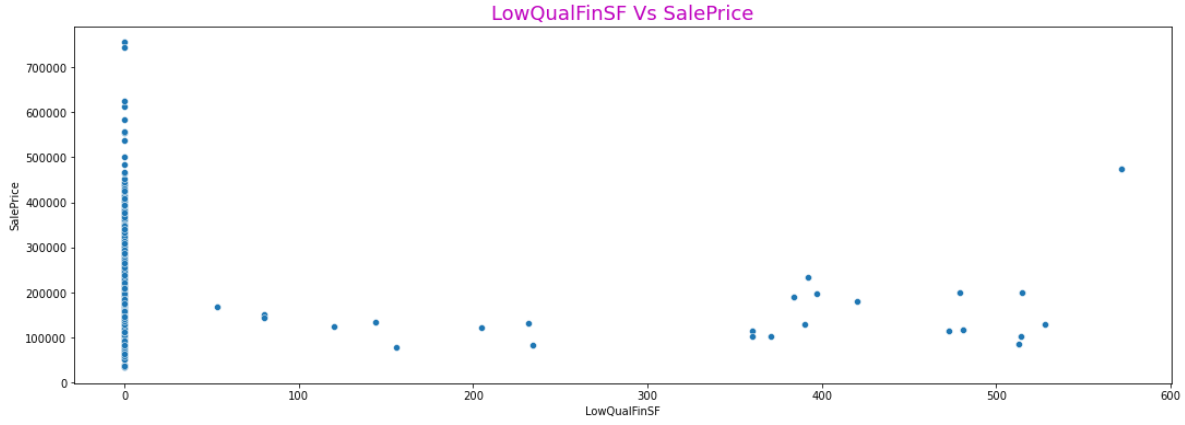
In [112]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="2ndFlrSF",y='SalePrice',data=train);
plt.title('2ndFlrSF Vs SalePrice',fontsize=18,color='m');
```



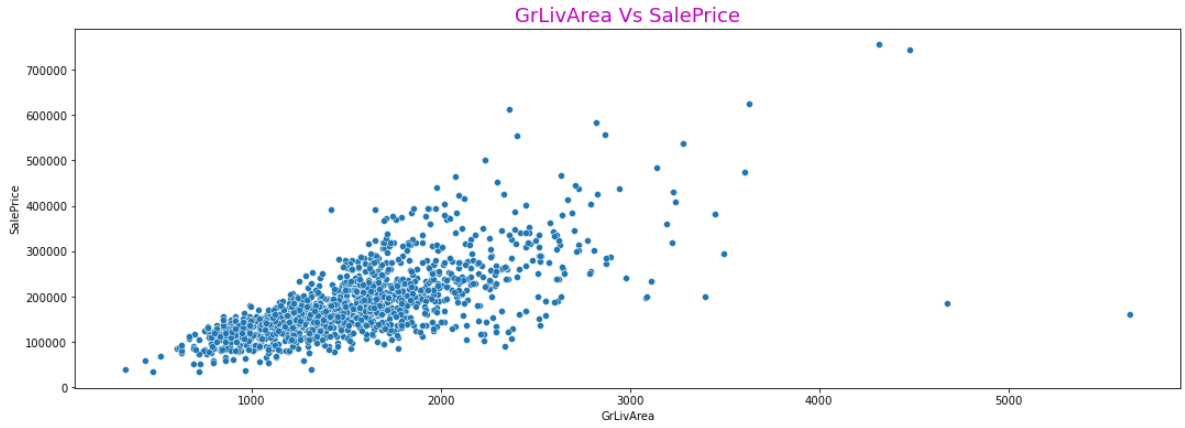
In [113]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="LowQualFinSF",y='SalePrice',data=train);
plt.title('LowQualFinSF Vs SalePrice',fontsize=18,color='m');
```



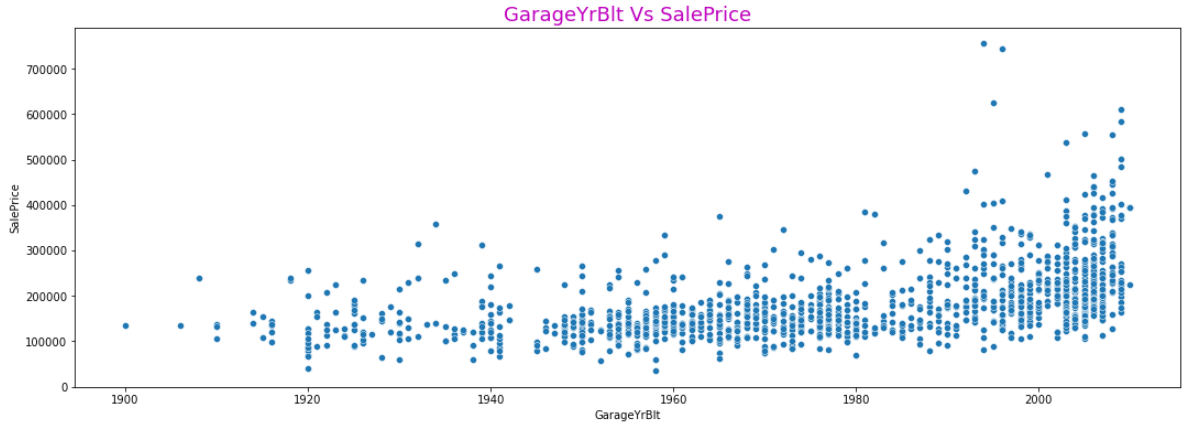
In [114]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="GrLivArea",y='SalePrice',data=train);
plt.title('GrLivArea Vs SalePrice',fontsize=18,color='m');
```



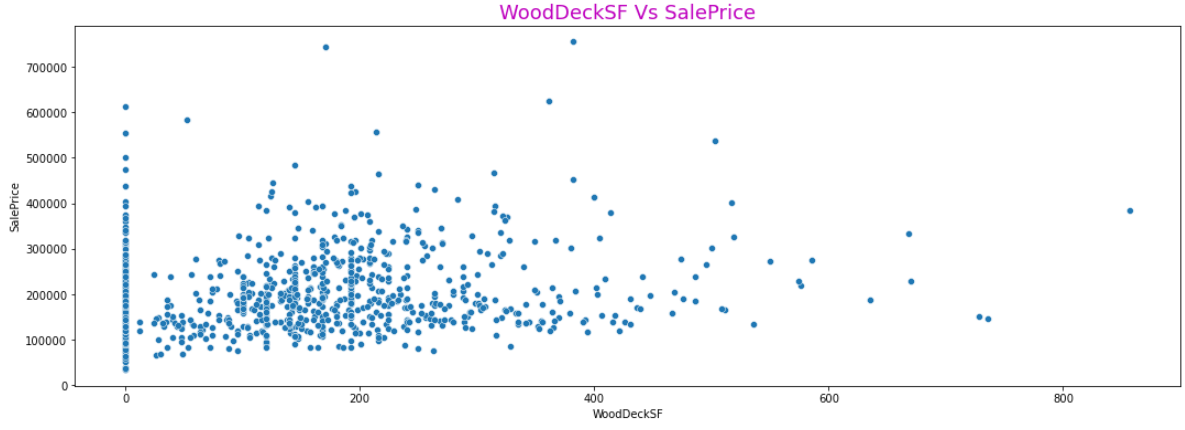
In [115]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="GarageYrBlt",y='SalePrice',data=train);
plt.title('GarageYrBlt Vs SalePrice',fontsize=18,color='m');
```



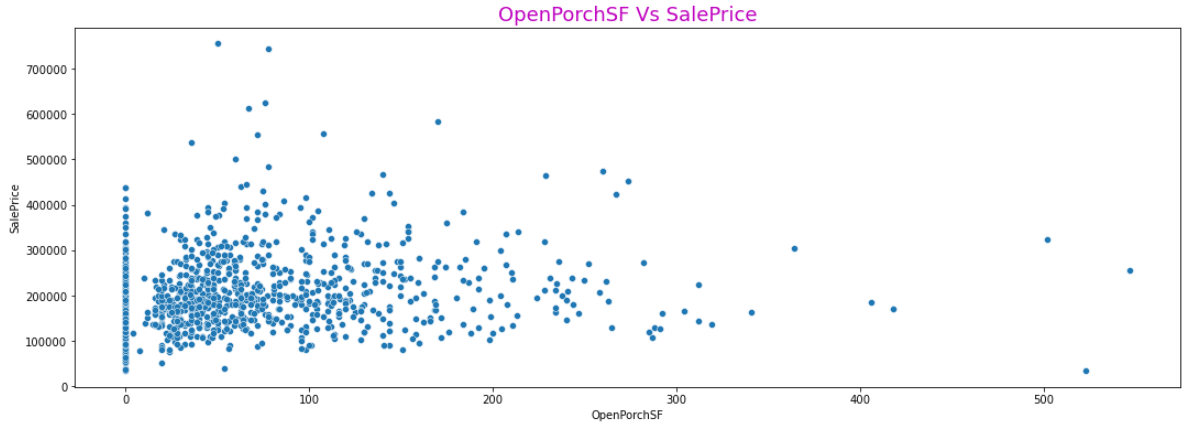
In [116]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="WoodDeckSF",y='SalePrice',data=train);
plt.title('WoodDeckSF Vs SalePrice',fontsize=18,color='m');
```



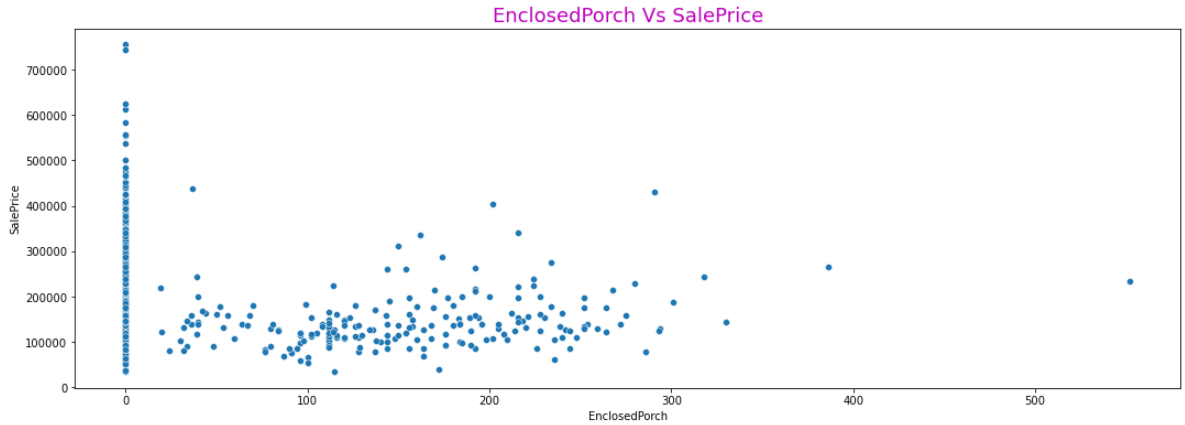
In [117]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="OpenPorchSF",y='SalePrice',data=train);
plt.title('OpenPorchSF Vs SalePrice',fontsize=18,color='m');
```



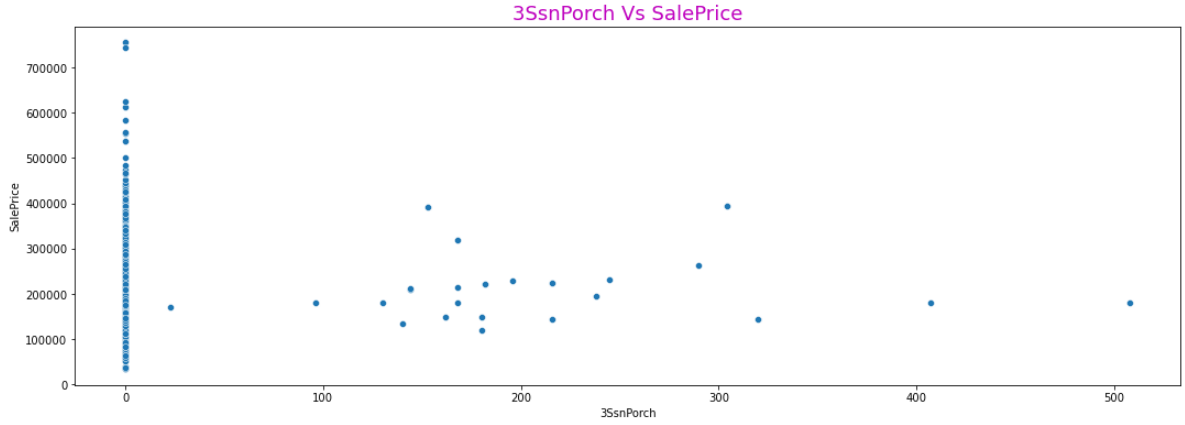
In [118]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="EnclosedPorch",y='SalePrice',data=train);
plt.title('EnclosedPorch Vs SalePrice',fontsize=18,color='m');
```



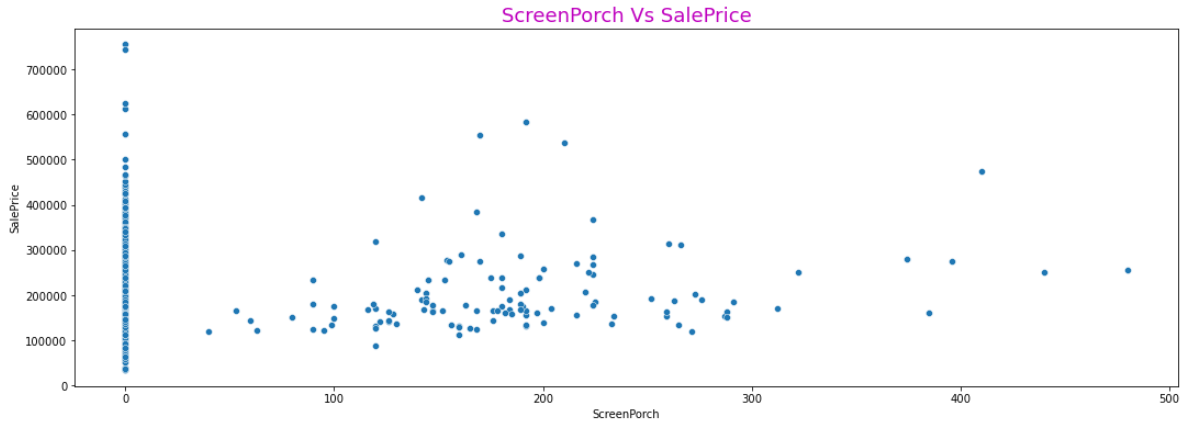
In [119]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="3SsnPorch",y='SalePrice',data=train);
plt.title('3SsnPorch Vs SalePrice',fontsize=18,color='m');
```



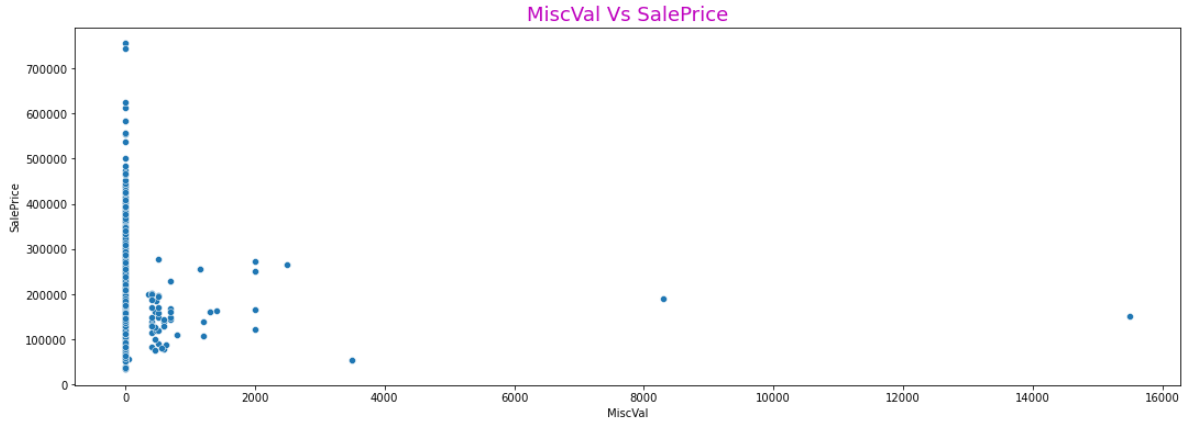
In [120]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="ScreenPorch",y='SalePrice',data=train);
plt.title('ScreenPorch Vs SalePrice',fontsize=18,color='m');
```



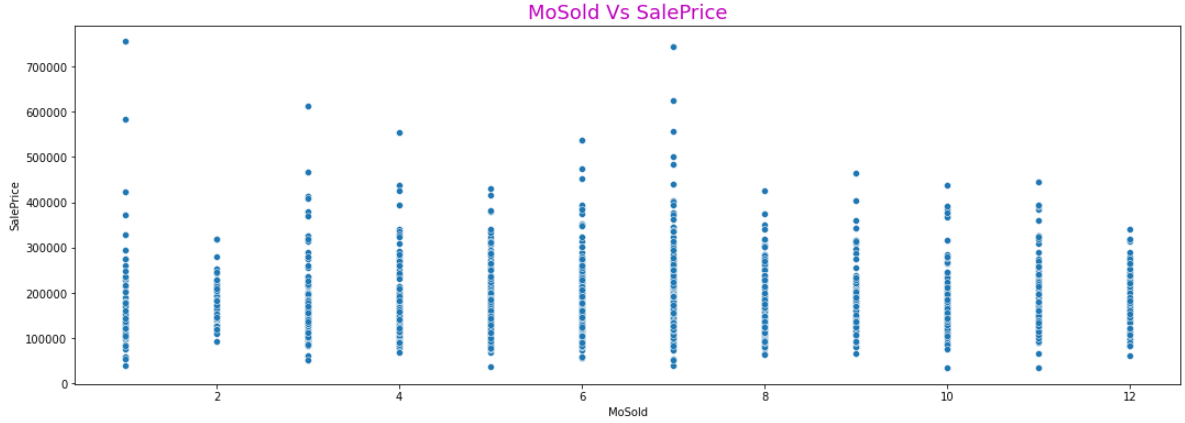
In [121]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="MiscVal",y='SalePrice',data=train);
plt.title('MiscVal Vs SalePrice',fontsize=18,color='m');
```



In [122]:

```
plt.figure(figsize=(18,6))
sb.scatterplot(x="MoSold",y='SalePrice',data=train);
plt.title('MoSold Vs SalePrice',fontsize=18,color='m');
```



In [ ]: