In [1]:

```
from sklearn.datasets import fetch_mldata
from sklearn.utils import shuffle
mnist = fetch_mldata("MNIST original")
X_digits, y_digits = mnist.data, mnist.target
X_digits, y_digits = shuffle(X_digits, y_digits)
```

In [2]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [3]:

```
X_digits_train = X_digits[:1000]
y_digits_train = y_digits[:1000]
X_digits_valid = X_digits[1000:2000]
y_digits_valid = y_digits[1000:2000]
X_digits_test = X_digits[2000:3000]
y_digits_test = y_digits[2000:3000]
```

In [9]:

```
#knn = KNeighborsClassifier(n_neighbors=3)
#knn.fit(X_digits_valid, y_digits_valid)
k_range = range(1, 15)

# We can create Python dictionary using [] or dict()
scores = []

# We use a loop through the range 1 to 15
# We append the scores in the dictionary
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_digits_train, y_digits_train)
    scores.append(knn.score(X_digits_test, y_digits_test)*100)

print(scores)
#print "Accuracy of KNN train set:", knn.score(X_digits_train, y_digits_train)*100,"%"
#print "Accuracy of KNN validation set:", knn.score(X_digits_valid, y_digits_valid)*100,"%"
#print "Accuracy of KNN test set:", knn.score(X_digits_test, y_digits_test)*100,"%"
#knn = KNeighborsClassifier(n_neighbors=3)
#knn.fit(X_digits_valid, y_digits_valid)
k_range = range(1, 15)

# We can create Python dictionary using [] or dict()
scores1 = []

# We use a loop through the range 1 to 15
# We append the scores in the dictionary
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_digits_valid, y_digits_valid)
    scores1.append(knn.score(X_digits_valid, y_digits_valid)*100)

print(scores1)
```

```
[87.0, 83.0, 86.299999999999997, 86.900000000000006, 86.5, 85.700000000000003, 86.5, 86.400000000000006
, 86.0, 85.700000000000003, 85.0, 85.599999999999994, 85.799999999999997, 85.200000000000003]
[100.0, 95.799999999999997, 95.700000000000003, 94.299999999999997, 93.600000000000009,
92.700000000000003, 92.5, 91.700000000000003, 91.299999999999997, 91.0, 90.400000000000006, 89.5, 89.40
0000000000006, 88.0]
```

In [10]:

```
print "Accuracy of KNN validation set:",max(scores1),"%"
print "Best number for k=n_neighbors is ",scores1.index(max(scores1))+1
print "Accuracy of KNN test set:",max(scores),"%"
print "Best number for k=n_neighbors is ",scores.index(max(scores))+1
```

```
Accuracy of KNN validation set: 100.0 %
Best number for k=n_neighbors is  1
Accuracy of KNN test set: 87.0 %
Best number for k=n_neighbors is  1
```

In [11]:

```
knnmodel = KNeighborsClassifier(n_neighbors=3)
knnmodel.fit(X_digits_train, y_digits_train)
predictions = knnmodel.predict(X_digits_test)
```

In [15]:

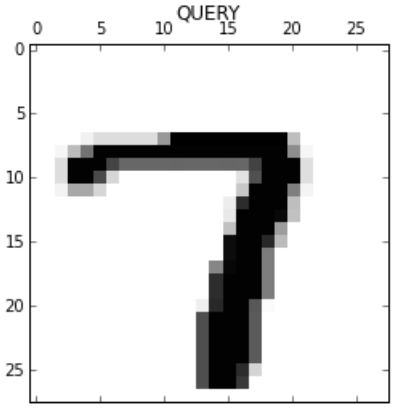
```
query=X_digits_test[y_digits_test==7]
prediction = knnmodel.predict(query)[0]
```

In [16]:

```
plt.rc("image", cmap="binary")

plt.matshow(query[0].reshape(28, 28))
plt.title('QUERY')
```

Out[16]:

```
<matplotlib.text.Text at 0xda363f0>
```

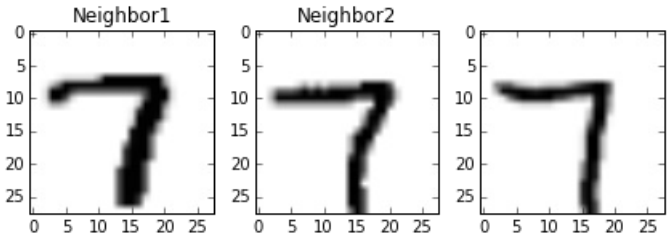


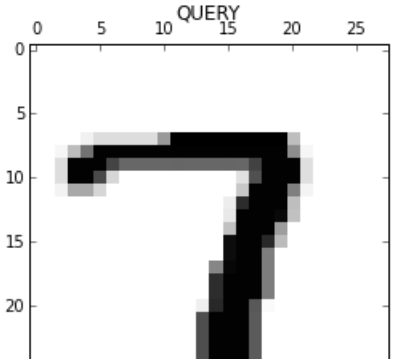
In [19]:

```
neighbor=X_digits_test[y_digits_test==prediction]
plt.rc("image", cmap="binary") # use black/white palette for plotting

for i in xrange(3):
    plt.title('Neighbor'+str(i))
    plt.subplot(2,3,i+1)
    plt.imshow(neighbor[i].reshape(28,28))

plt.tight_layout()
plt.matshow(query[0].reshape(28, 28))
plt.title('QUERY')
```

Out[19]:

```
<matplotlib.text.Text at 0x1706aa90>
```



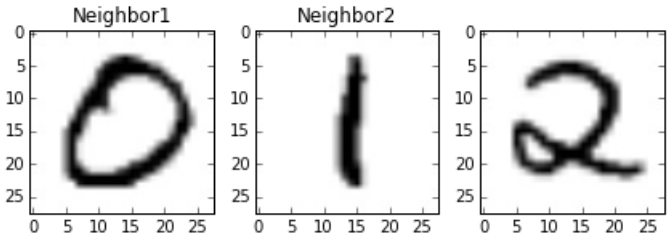


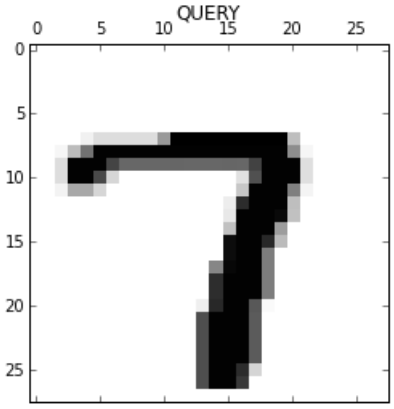
In [21]:

```
neighbor2=X_digits_test[y_digits_test!=prediction]
plt.rc("image", cmap="binary") # use black/white palette for plotting
for i in xrange(3):
    plt.title('Neighbor'+str(i))
    plt.subplot(2,3,i+1)
    plt.imshow(neighbor2[i].reshape(28,28))
plt.tight_layout()
plt.matshow(query[0].reshape(28, 28))
plt.title('QUERY')
```

Out[21]:

```
<matplotlib.text.Text at 0xda1b210>
```





In [47]:

```
nan
```

In [25]:

```

```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-25-ae1a05a2d7e1> in <module>()
----> 1 index(neighbor2)

NameError: name 'index' is not defined
```

In [ ]:

```

```