1.1 Setting Up OpenSSL

Before installing OpenSSL a Java Development kit (jdk) has to be installed.
OpenSSL is downloaded from its website and the .exe is clicked to open Open SLL terminal window. After this, the command below entered to create **admin-private-key.pem** file:

```
OpenSSL> req -x509 -newkey rsa:2048 -config "E:\OpenSSL\openssl-0.9.8e_X64\opens
sl.cnf" -keyout admin-private-key.pem -out admin-cert.pem -days 365 -subj "/CN=A
dmin Q. User/C=US/L=Seattle" -nodes
Loading 'screen' into random state - done
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'admin-private-key.pem'
-----
```

To create **admin-cert.pem** and **admin-q-user.pfx :**

```
-----
OpenSSL> pkcs12 -inkey admin-private-key.pem -in admin-cert.pem -export -out adm
in-q-user.pfx -passout pass:"SuperSecret"
Loading 'screen' into random state - done
```

To create keystore, the command below entered:

```
OpenSSL> keytool -genkeypair -alias nifiserver -keyalg RSA -keypass SuperSecret -storepass SuperSecret -keystore server_
keystore.jks -dname "CN=Test NiFi Server" -noprompt
```

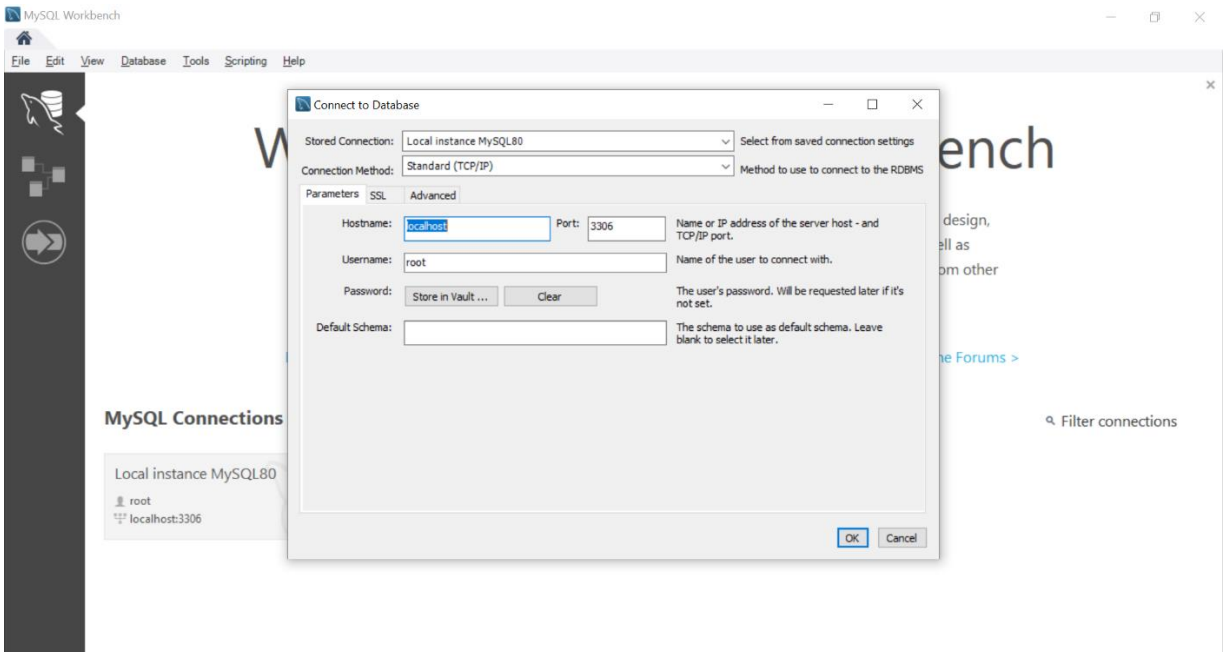
To add SSL certificate to the KeyStore, the below command is used:

```
C:\Program Files\Java\jdk1.8.0_121\bin>
C:\Program Files\Java\jdk1.8.0_121\bin>keytool -importcert -v -trustcacerts -ali
as admin -file E:\OpenSSL\openssl-0.9.8e_X64\bin\admin-cert.pem -keystore server
_keystore.jks  -storepass SuperSecret -noprompt
Certificate was added to keystore
[Storing server_keystore.jks]
```

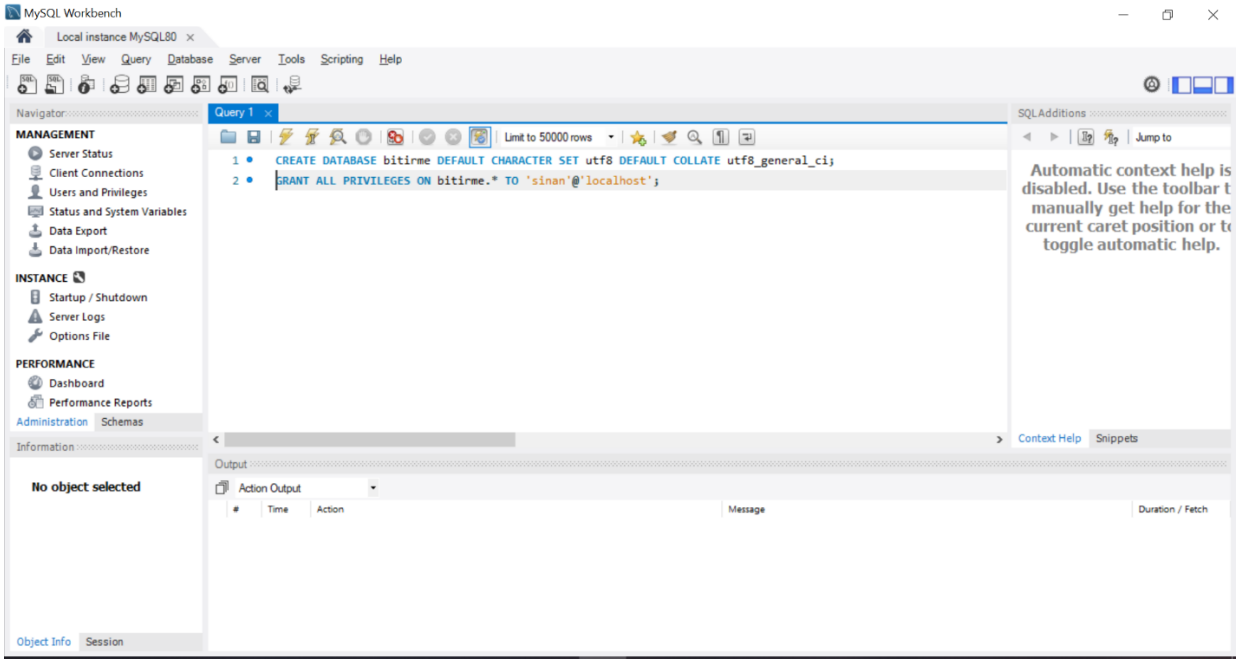
Normally a trust store is also needed for establishing HTTPS connection but in this project, Java Development Kit's own trust store used.

1.2 Creating Mysql Database

Creating for MySql Database, first the set up file downloaded from its official web page. Community Edition selected for being free of charge. After downloading set-up files, .exe file clicked and a server application, client application, an ide (MySql Workbench) installed by following directions. A root user and 'sinan' user ,which is for client connection, defined for database administration. After these steps, MySql Workbench is opened, and a connection established with "root" user:



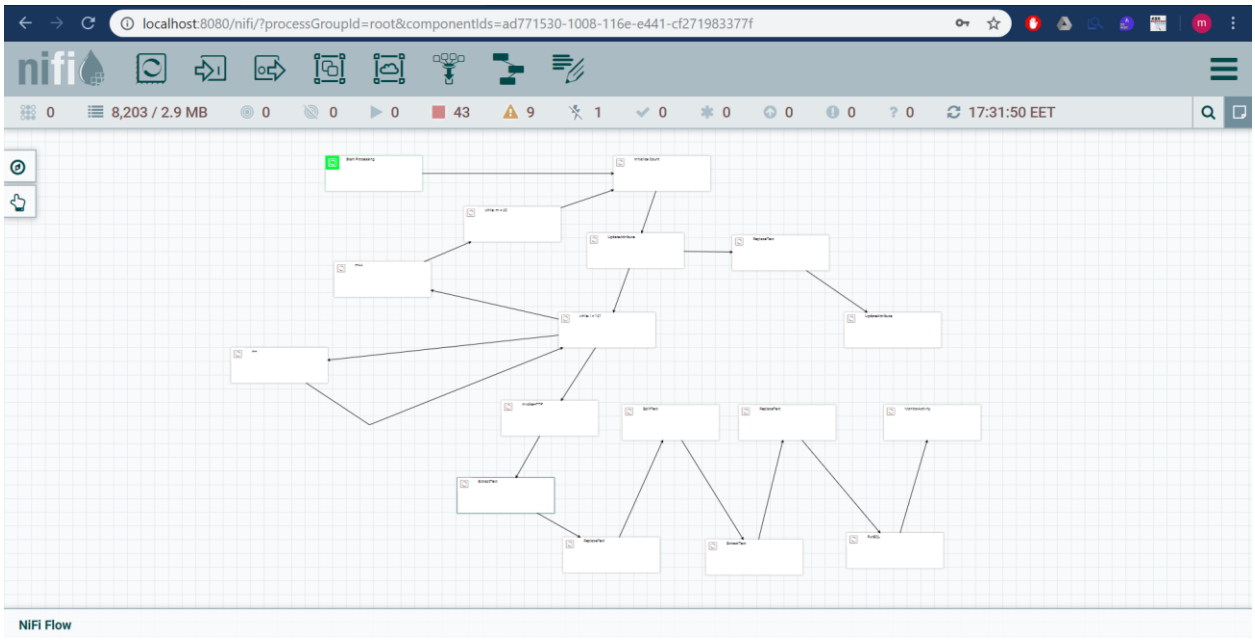
After this, a database created, which is named ‘bitirme’ and user ‘sinan’ granted for database created. Default character set defined as ‘UTF-8’ in order to avoid Turkish character problems:



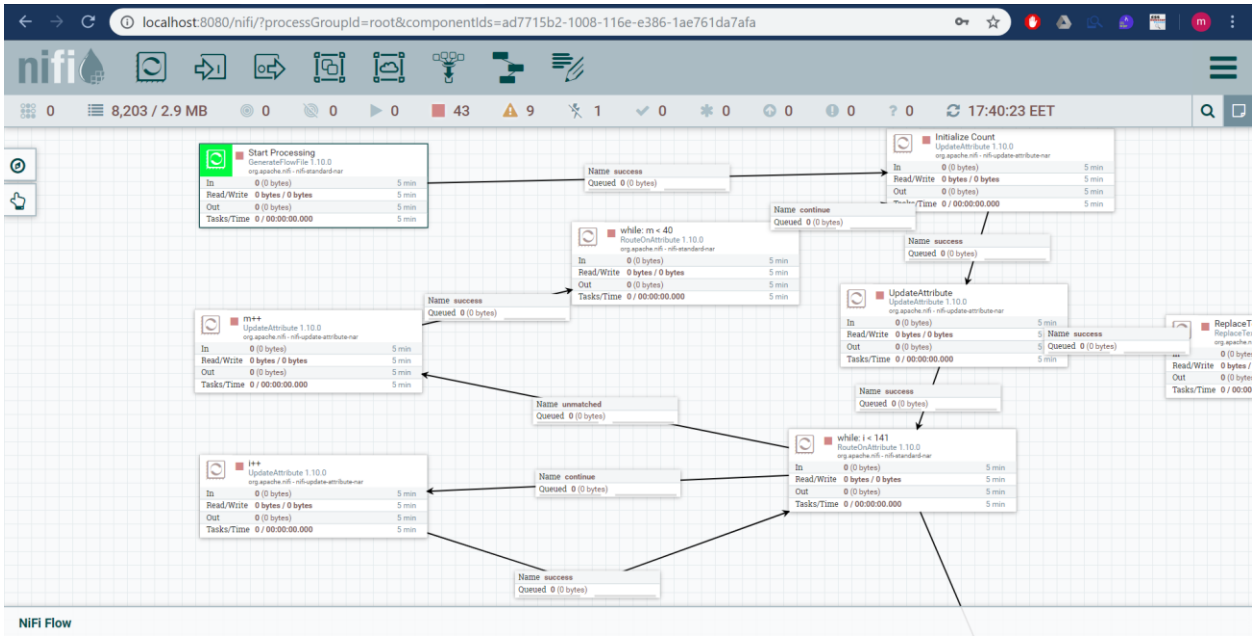
1.3 Setting up NiFi for Web Crawling

Apache NiFi is open source data integration/ETL program. It is downloaded from its official web site. Nifi does not requires any installation process for standalone use. Clicking ‘run.bat’ starts nifi service in a few mininutes. To open NiFi user interface, ‘localhost:8080/nifi’ has to

be written on any browser. All processor and their connections for data crawling is shown below:

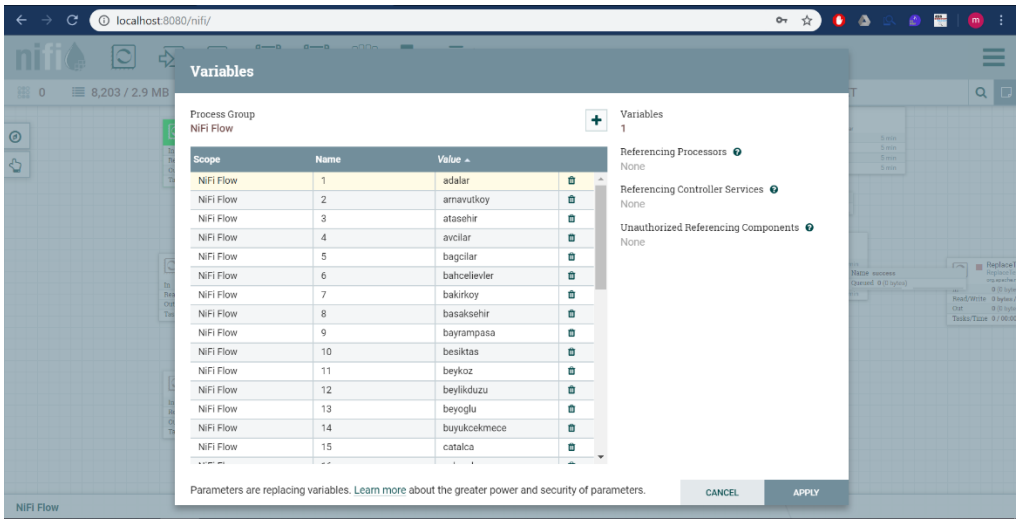


Creating dynamic web link for crawling all data for İstanbul is accomplished by these processors:

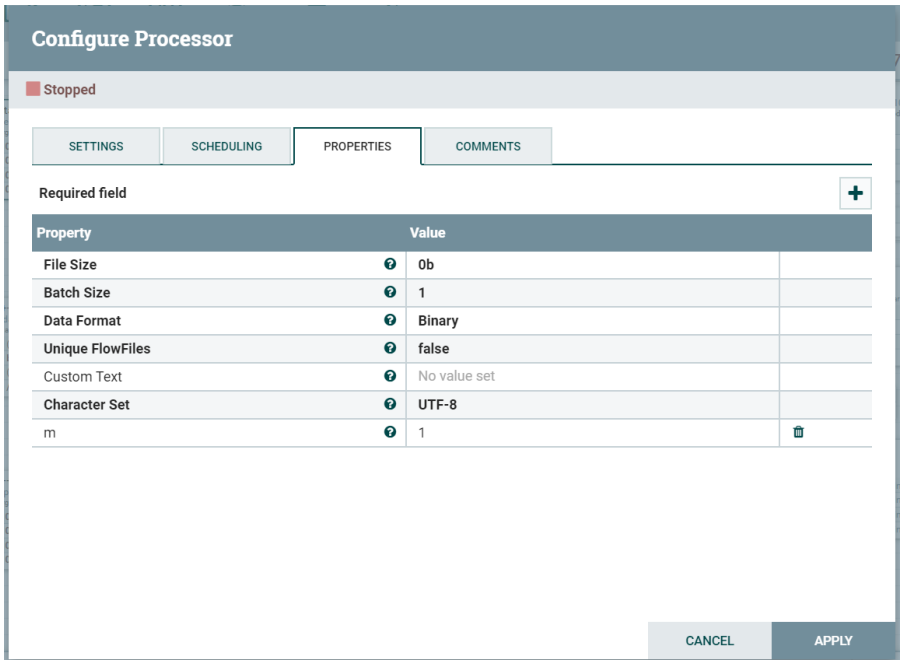


In hürriyetemlak.com, there is page limitation for every selected content as ‘140’ (when showing 50 result per page) So if the link which contains all districts in İstanbul selected and

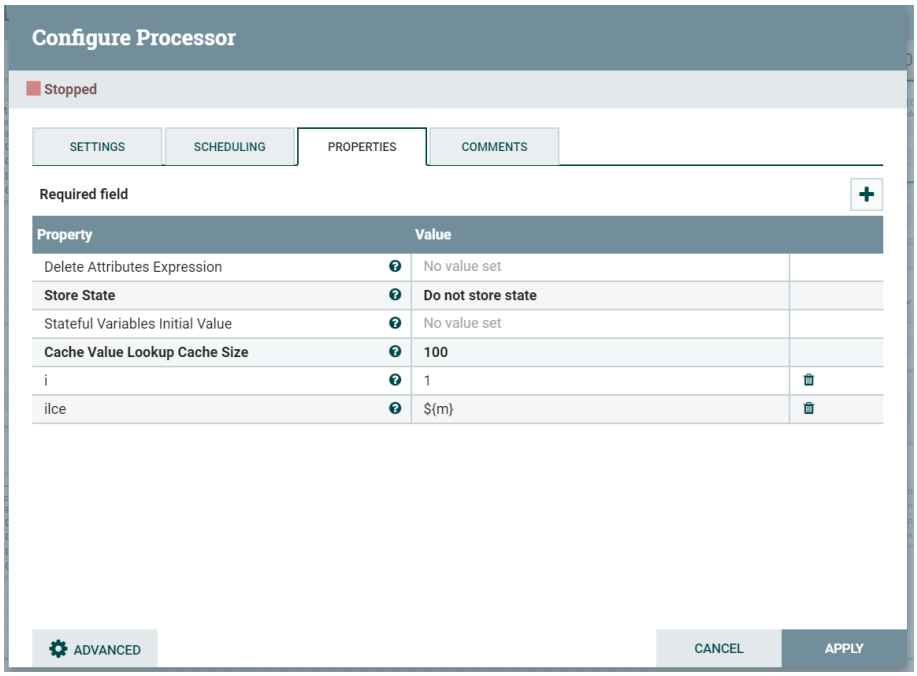
crawled, only 7000 real estate info could be downloaded despite having 60.000. Because of this dynamic link has to be created from district links. This is all İstanbul link: "https://www.hurriyetemlak.com/istanbul-satilik?pageSize=50" and this is the link for Adalar district :" https://www.hurriyetemlak.com/adalar-satilik?pageSize=50". To create link automatically and look for all 140 pages for all district, two loop are designed. The inner loop changes page number name and outer loop changes district name. All district names are defined in NiFi as variable first:



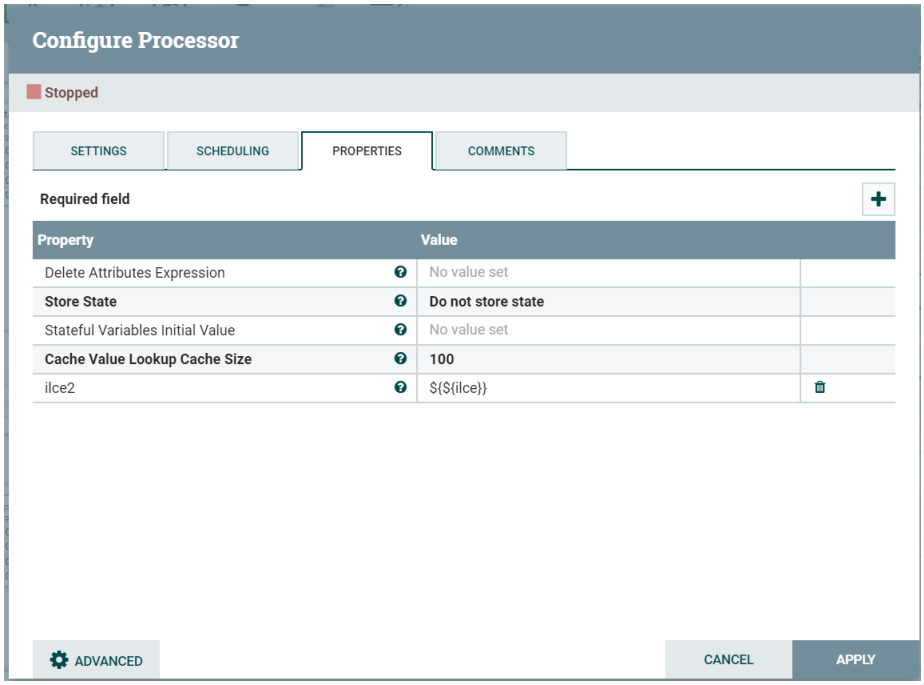
After these definitions, firs processor added to trigger all process and defining 'm' value. 'm' will be used creating dynamic district name later.



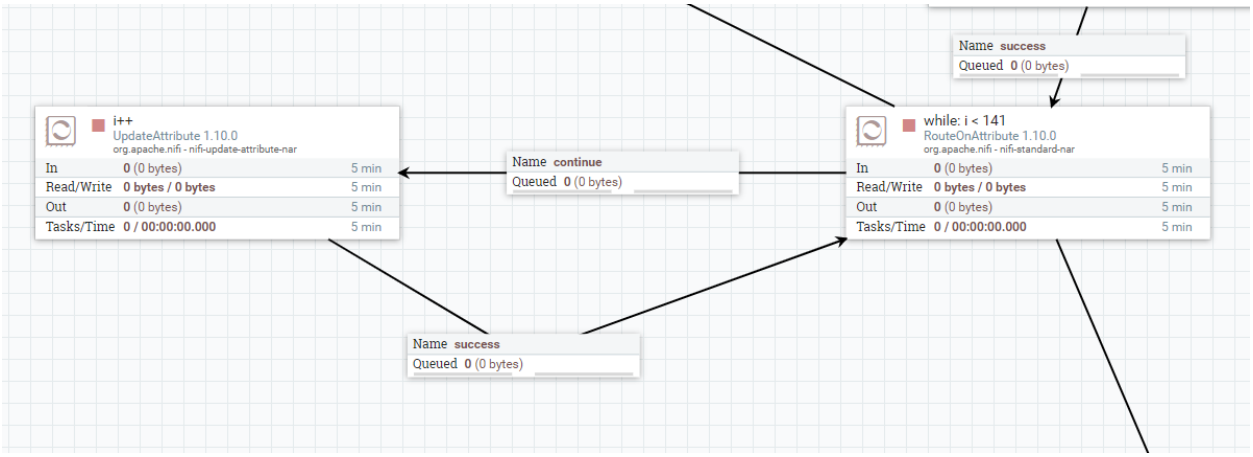
‘Initialize Count’ is the processor which is used for defining ‘ilce’ value based on ‘m’ value and ‘I’ value which is created for dynamic page number.



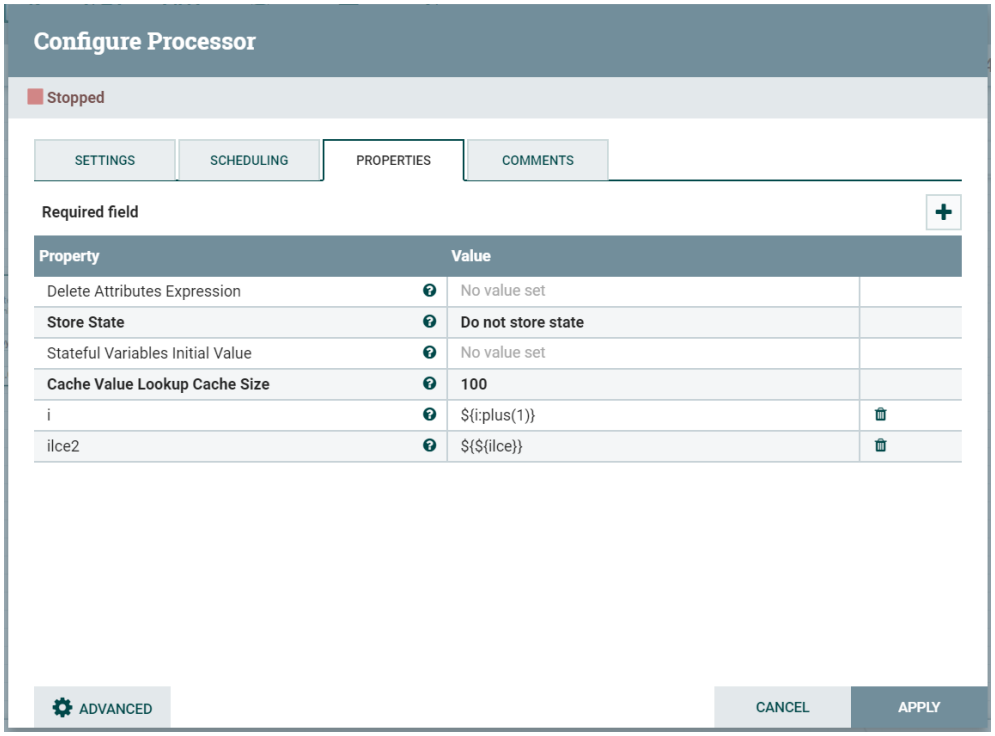
‘Ilce’ variable created from ‘m’ variable and dynamic link which includes this variable is created using ‘UpdateAttiribute’ processor.



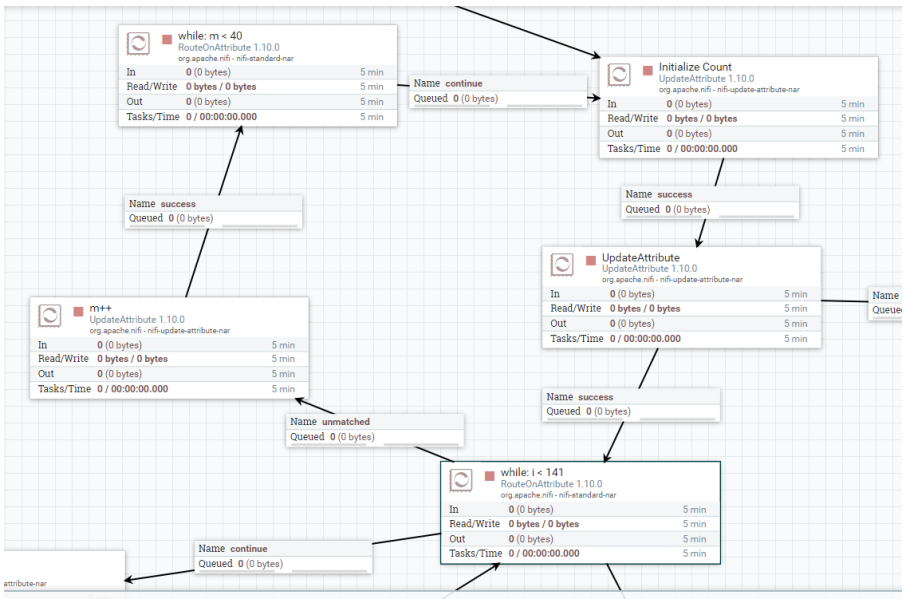
The next processor is ‘while: i < 141’ it checks the value of ‘I’ (page number variable) if it is smaller then 141, it routes the flow inner loop to increasing page number variable one by one. If ‘I’value is greater then 140 then it routes the flow outer root to update district variables (m,ilce,ilce2). The inner loop:



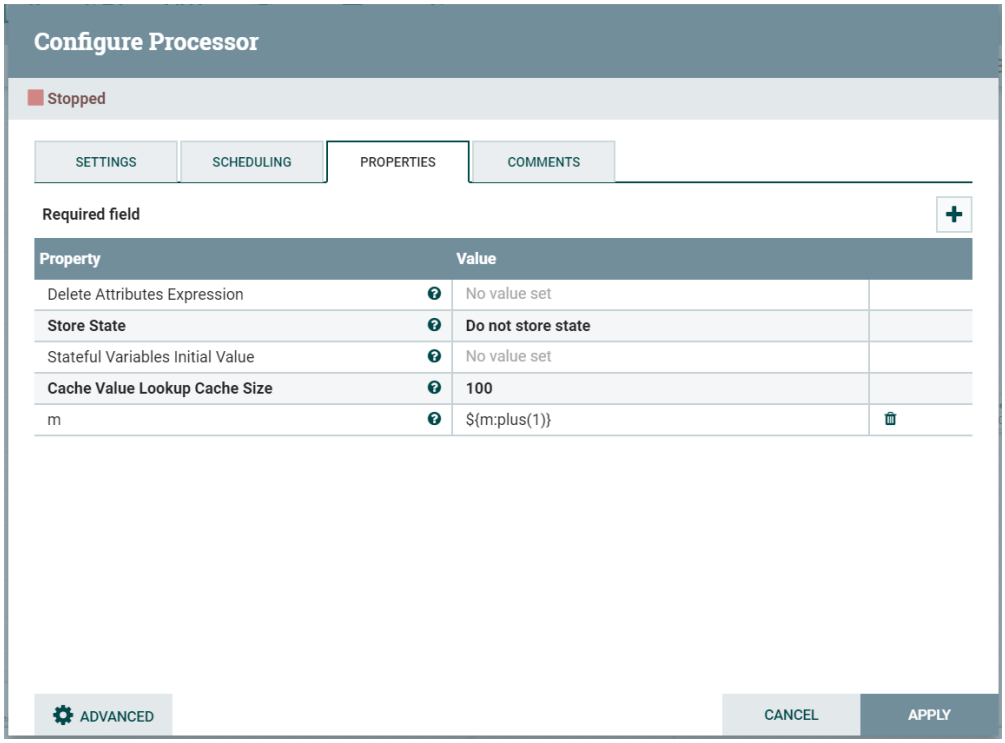
The processor which is named as ‘i++’ increases page number variable adding 1 every time it works. ‘ilce2’ variable is also defined here because of ‘InvokeHTTP’ processor needs this flow attributes every time it works. This variable just keeps and passed district value which comes from outer loop.



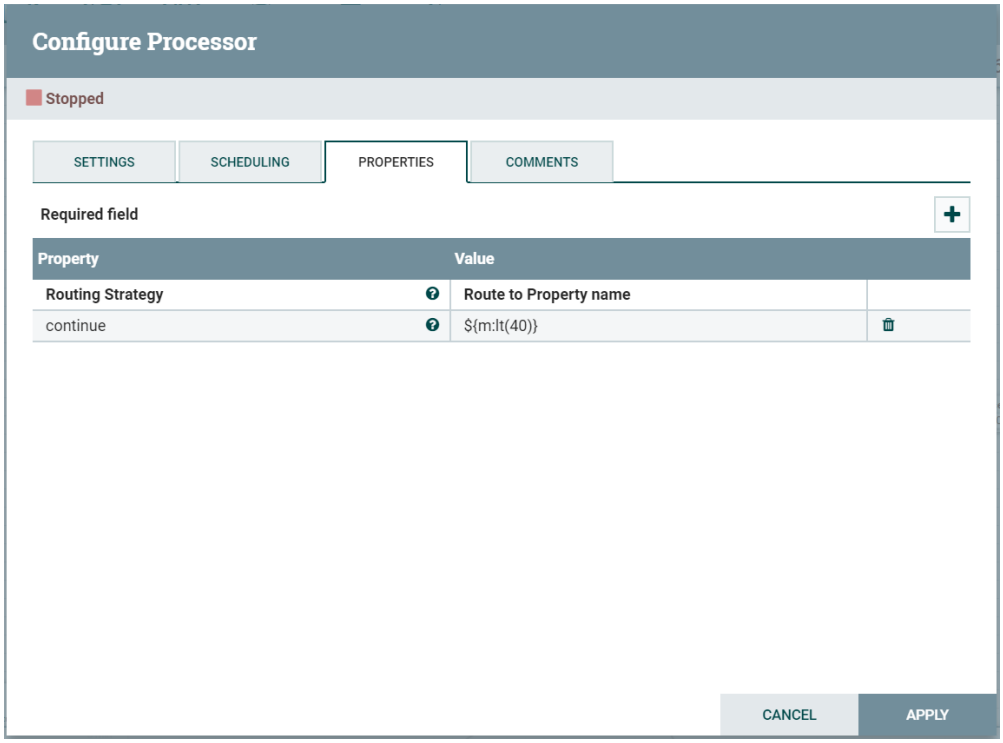
The outer loop,which is shown below, increases ‘m’ variable by adding one and changes district value in the flow. There are 39 district in İstanbul so when ‘m’ value reaches ‘40’ it cuts the flow and stops all data flow.



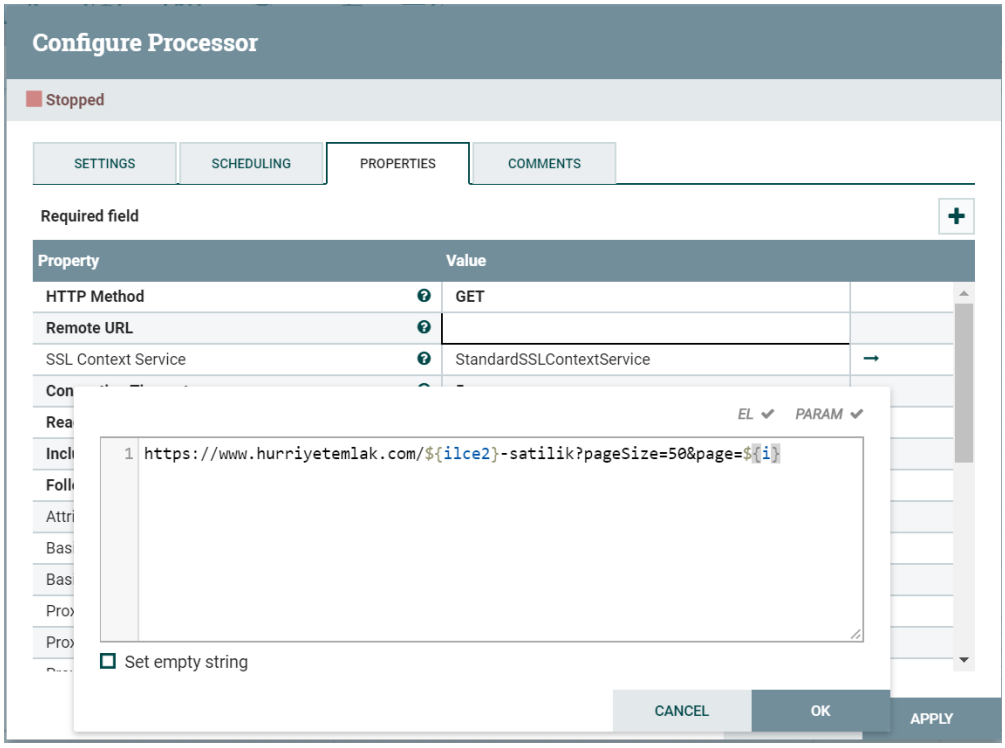
Properties of the ‘m++’ processor is shown below:



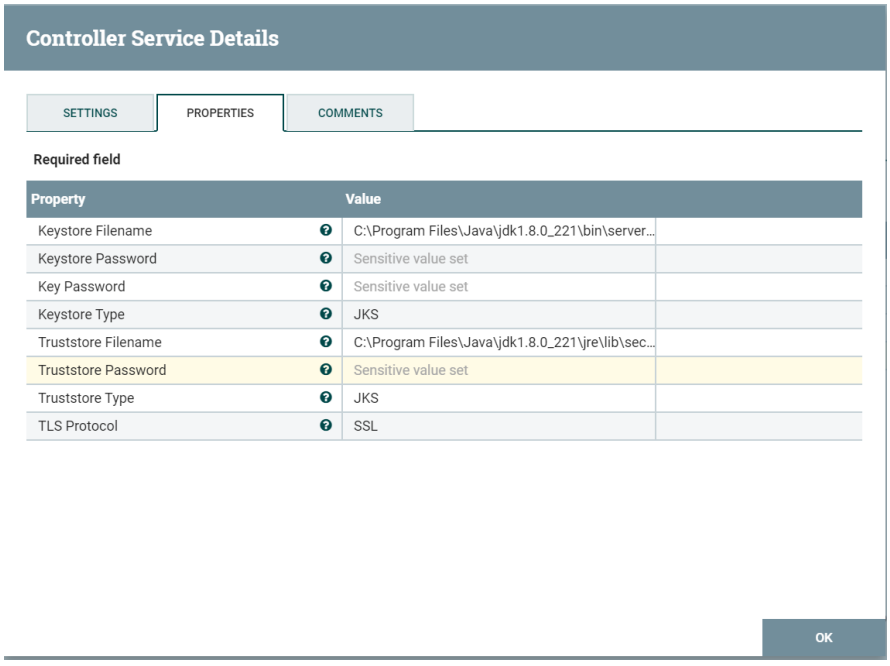
Properties of the ‘while: m < 40’ processor is shown below:



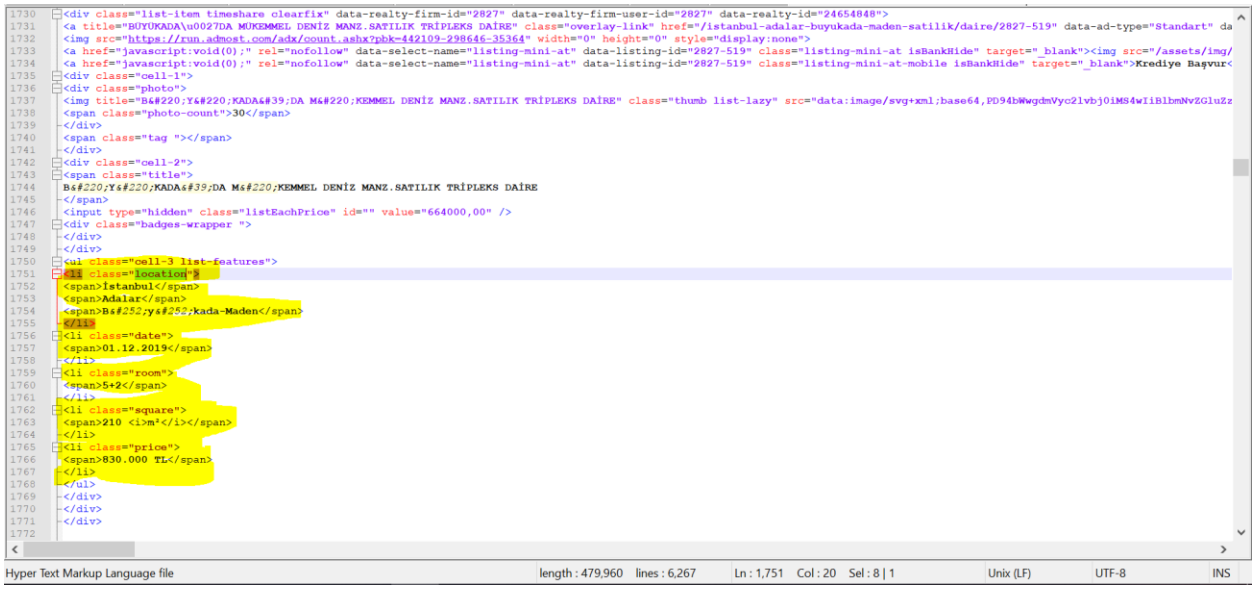
The next processor on the flow is ‘InvokeHTTP’ processor. It is selected to download html file of hürriyetemlak.com. It takes flowfile content before itself and uses “get” method to download web content.



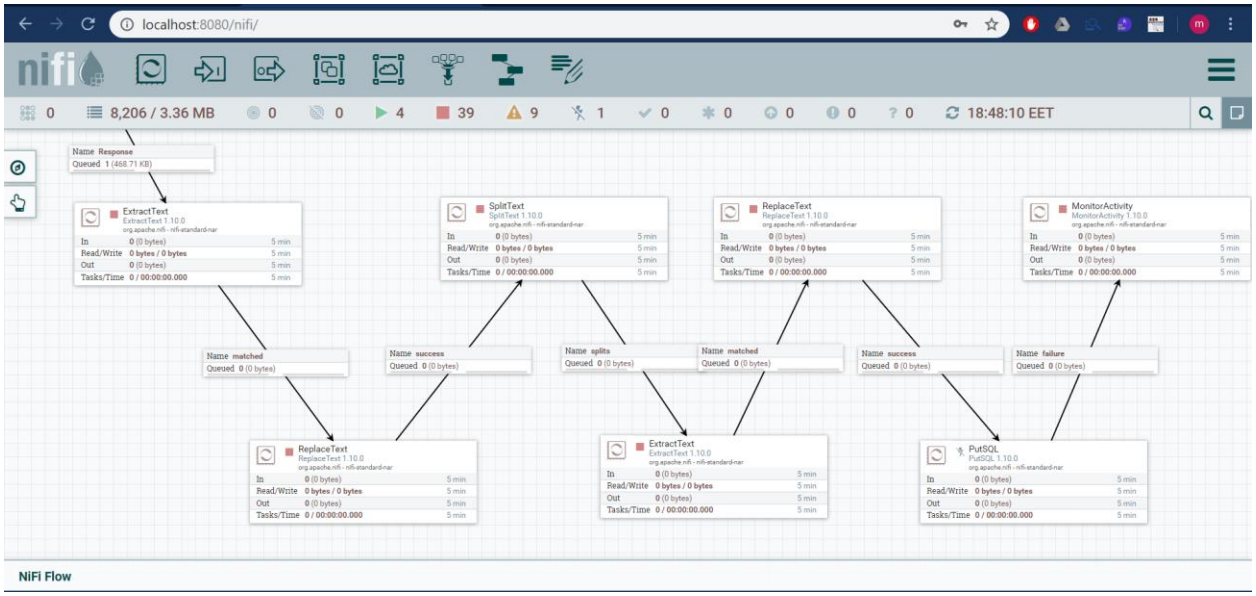
To use ‘invokeHTTP’ processor with HTTPS links, ‘SSL Content Service’ has to be created with OpenSSL values which created before.



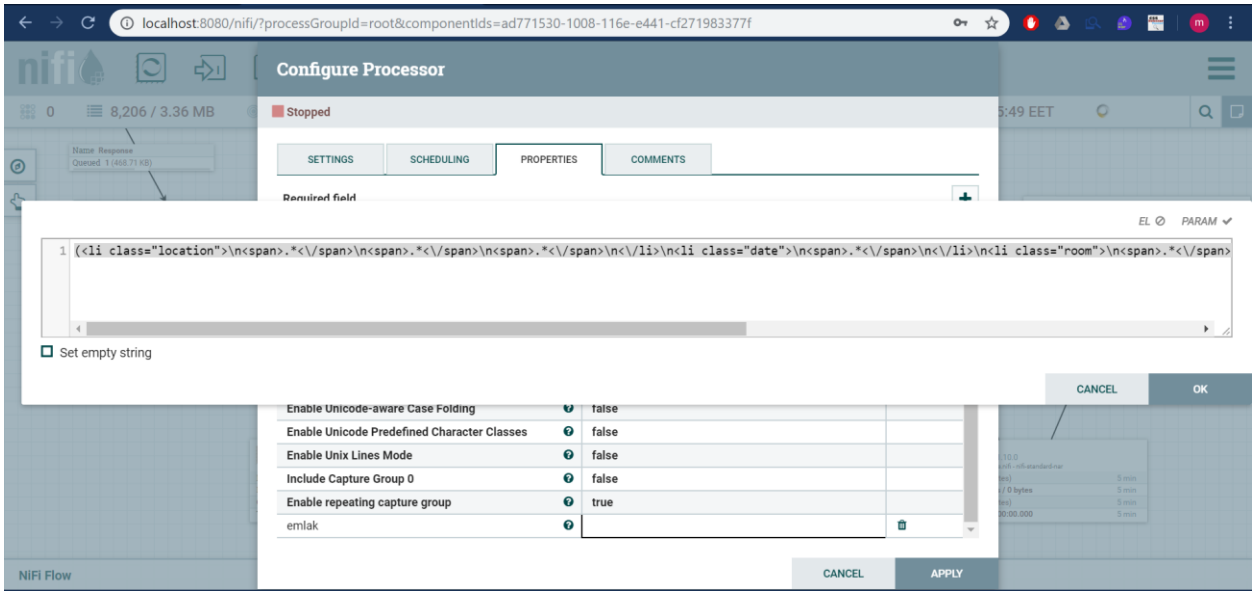
When the processors above started, a html file derived from hürriyetemlak.com as shown below.



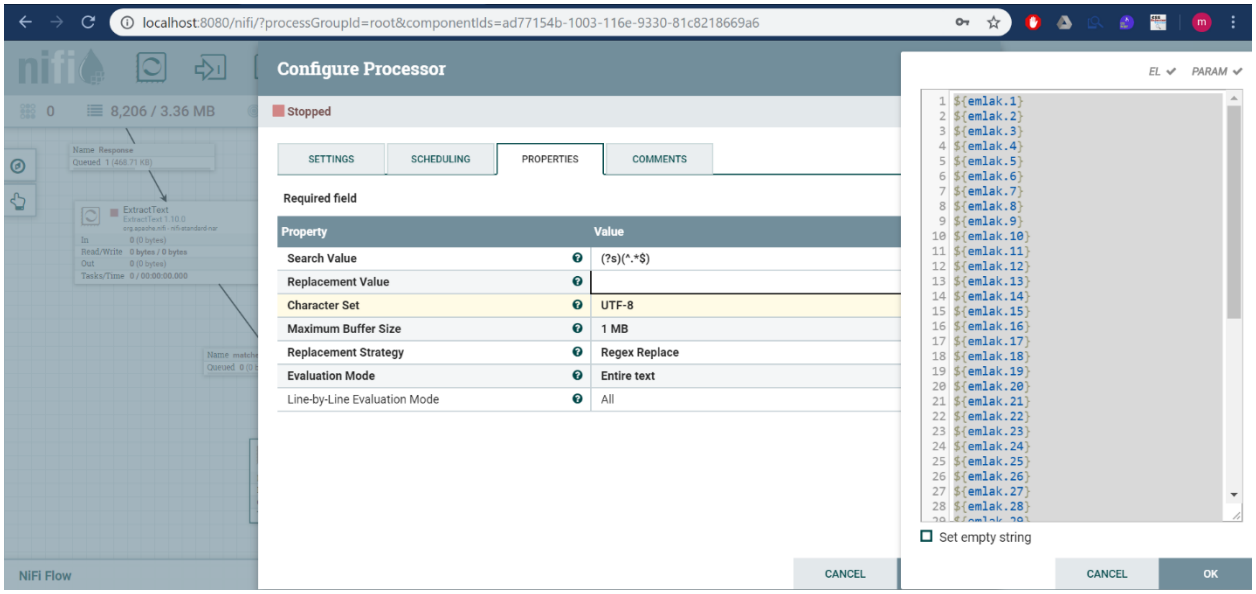
After getting this html file, data tranfomation process begins to extact real estate information from HTML file. The processor, which are doing data transformation, are shown below:



First processor in data transformation side is ‘ExtractText’ processor. It matches text pattern using regular expression and defines it a flowfile attribute which is manually created. For this case, manually created flowfile attribute is ‘emlak’.



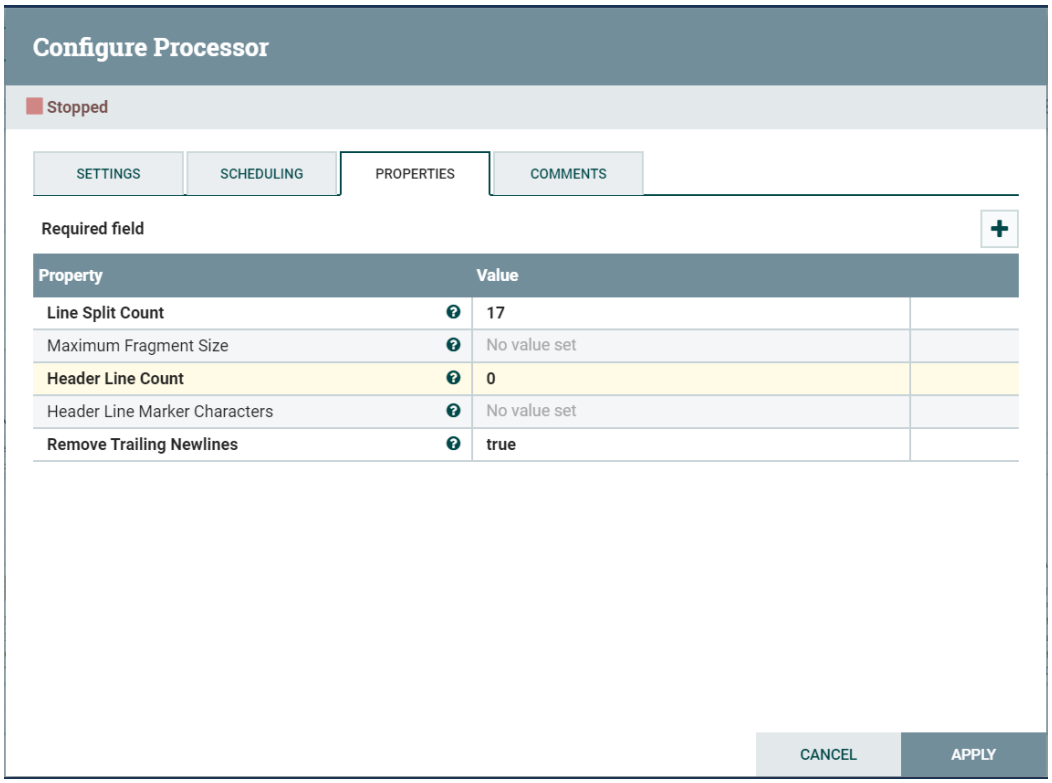
The next processor is ‘ReplaceText’ processor. It is used replace all flow file content with ‘emlak’ content. All emlak content has and index which represent capturing group of regular expression. There are 50 real estate data on every page so there is 50 capturing groups.



After first ‘ReplaceText’ processor, html data tranforms this form:

```
<li class="location">
 <span>İstanbul</span>
 <span>Adalar</span>
 <span>B&#252;y&#252;kada-Maden</span>
 </li>
<li class="date">
 <span>01.12.2019</span>
 </li>
<li class="room">
 <span>5+2</span>
 </li>
<li class="square">
 <span>210 <i>m²</i></span>
 </li>
<li class="price">
 <span>830.000 TL</span>
 </li>
<li class="location">
 <span>İstanbul</span>
 <span>Adalar</span>
 <span>B&#252;y&#252;kada-Nizam</span>
 </li>
<li class="date">
 <span>01.12.2019</span>
 </li>
<li class="room">
 <span>3+2</span>
 </li>
<li class="square">
 <span>290 <i>m²</i></span>
 </li>
<li class="price">
 <span>1.985.000 TL</span>
 </li>
<li class="location">
 <span>İstanbul</span>
 <span>Adalar</span>
 <span>B&#252;y&#252;kada-Maden</span>
 </li>
<li class="date">
 <span>01.12.2019</span>
 </li>
<li class="room">
 <span>2+1</span>
```

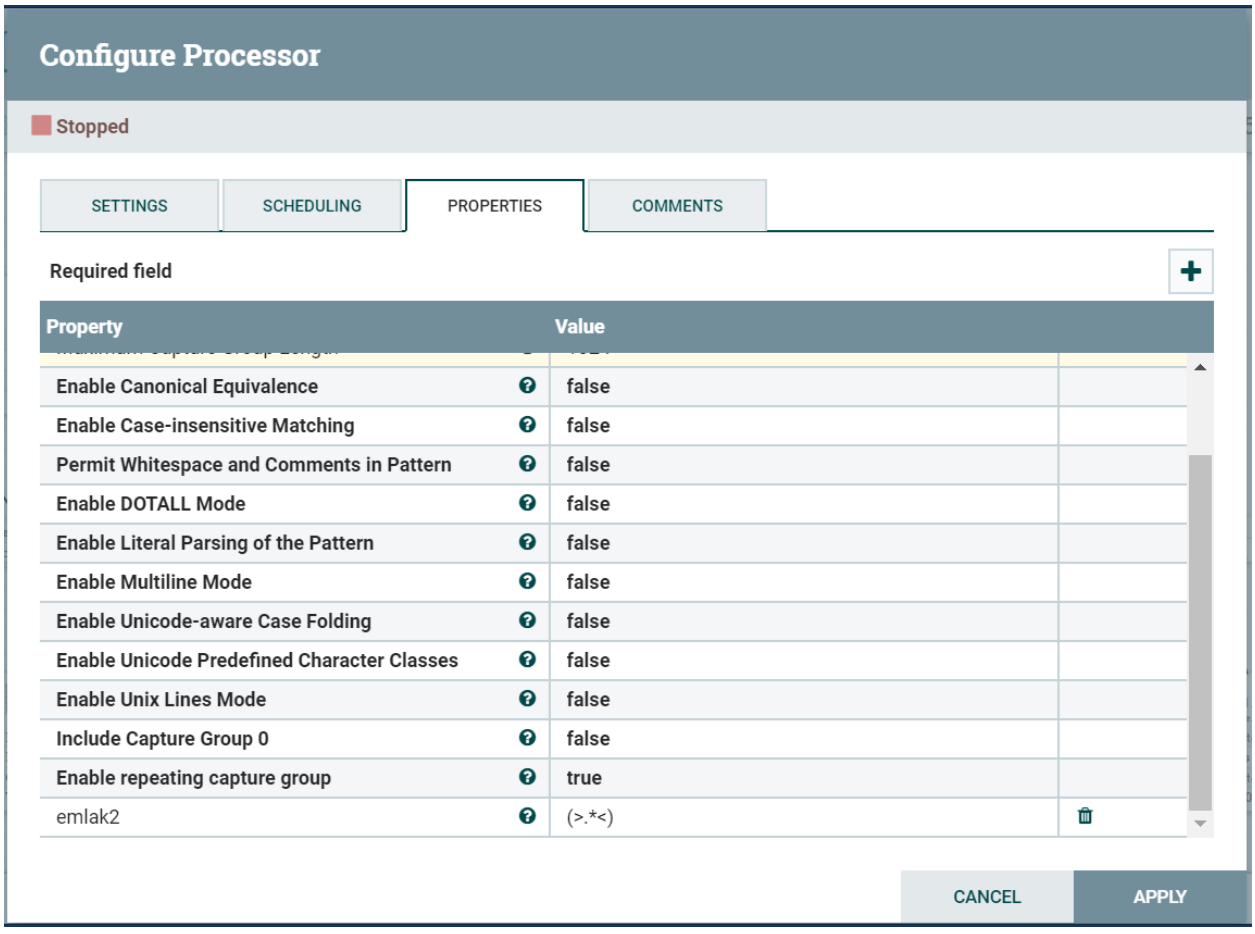
‘SplitText’ processor splits text by line numbers and separates code blocks to process every real estate data individually. Thanks to this processor, not only processing code block one by one decreases CPU usage, but also it will be very handy while creating insert script on next processes.



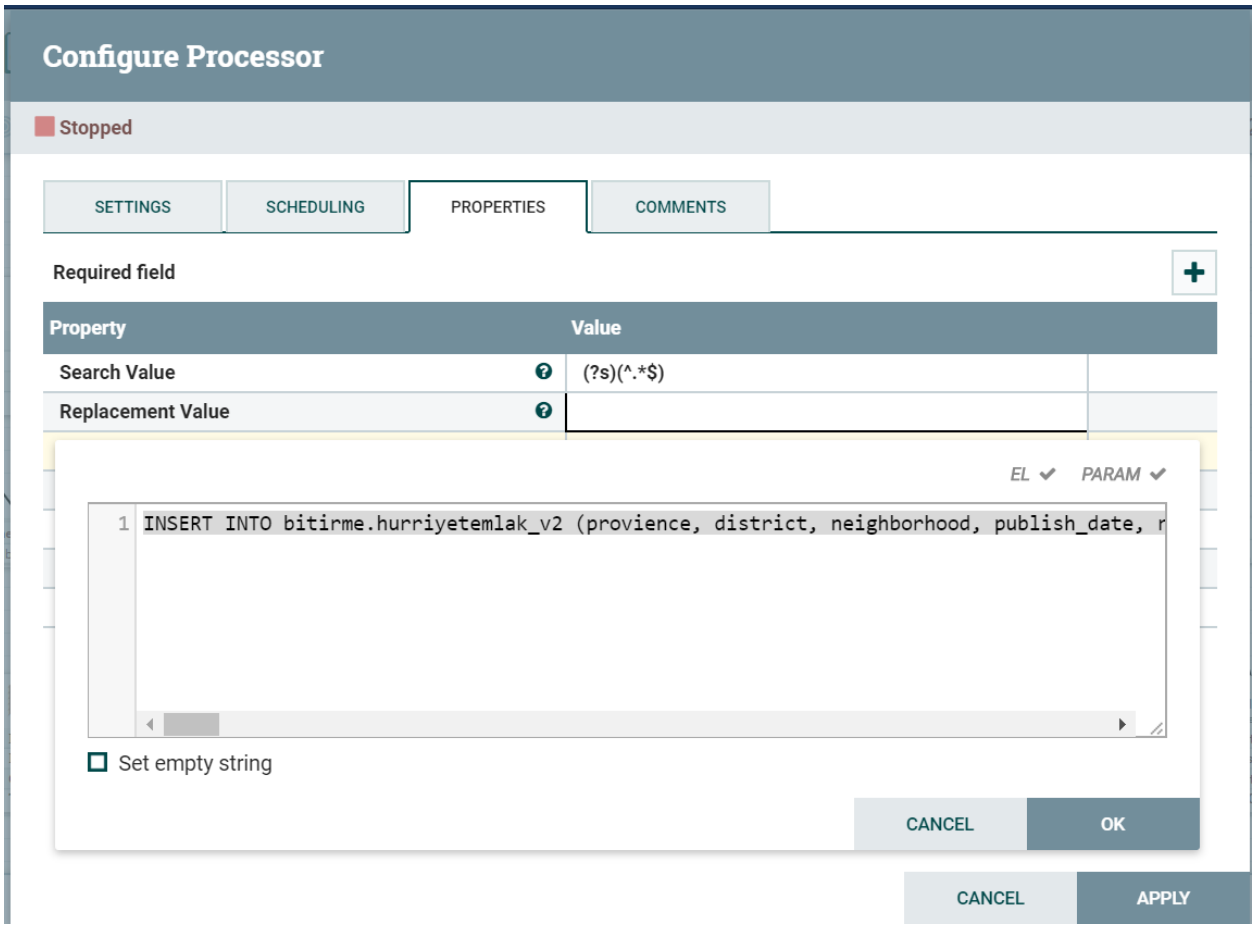
After this process, flow file data is divided 50 different code blocks. Example code block:

```
<li class="location">
<span>İstanbul</span>
<span>Adalar</span>
<span>B&#252;y&#252;kada-Maden</span>
</li>
<li class="date">
<span>01.12.2019</span>
</li>
<li class="room">
<span>5+2</span>
</li>
<li class="square">
<span>210 <i>m²</i></span>
</li>
<li class="price">
<span>830.000 TL</span>
</li>
```

The next processor is ‘ExtractText’ processor again. This time it is used to extract attributes from code blocks. To match patterns, regular expressions are used. The new manual flow file attribute name is ‘emlak2’.



After that, ‘ReplaceText’ processor used again. Using regular expression capture groups, a sql insert script created from each code blocks. In addition to this, Turkish characters matched with Unicode values and replaced English characters. In fact, NiFi is set to ‘UTF-8’ format bu ‘ReplaceText’ processor has an bug an it spoils character set.



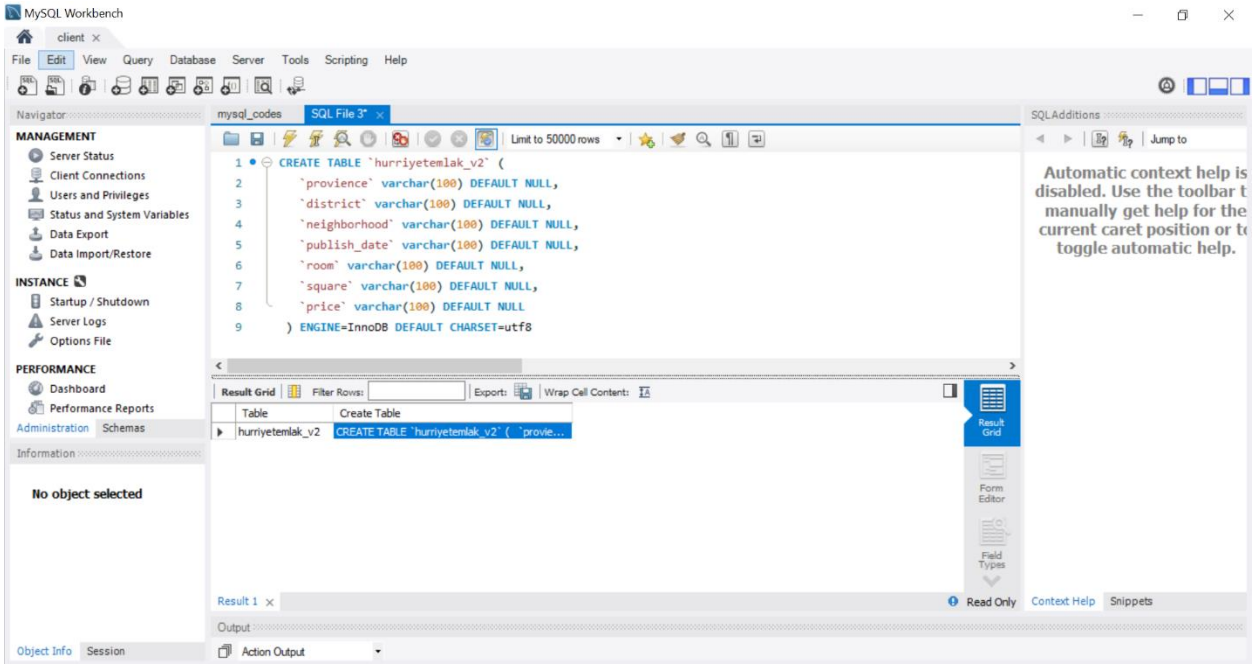
Full code of replacement string:

```
INSERT INTO bitirme.hurriyetemlak_v2 (provience, district, neighborhood, publish_date, room, square, price)
VALUES ("${emlak2.1:replaceAll("<|>",""):replace("&#304;","I"):replace("&#286;","G"):replace("&#350;","S"):
replace("&#199;","C"):replace("&#220;","U"):replace("&#214;","O"):replace("&#305;","i"):replace("&#287;","g"):
replace("&#351;","s"):replace("&#231;","c"):replace("&#252;","u"):replace("&#246;","o")}","${emlak2.2:
replaceAll("<|>",""):replace("&#304;","I"):replace("&#286;","G"):replace("&#350;","S"):replace("&#199;","C"):
replace("&#220;","U"):replace("&#214;","O"):replace("&#305;","i"):replace("&#287;","g"):replace("&#351;","s"):
replace("&#231;","c"):replace("&#252;","u"):replace("&#246;","o")}","${emlak2.3:replaceAll("<|>",""):
replace("&#304;","I"):replace("&#286;","G"):replace("&#350;","S"):replace("&#199;","C"):replace("&#220;","U"):
replace("&#214;","O"):replace("&#305;","i"):replace("&#287;","g"):replace("&#351;","s"):replace("&#231;","c"):
replace("&#252;","u"):replace("&#246;","o")}","${emlak2.4:replaceAll("<|>","")}","${emlak2.5:replaceAll("<|>",""):
replace("&#304;","I"):replace("&#286;","G"):replace("&#350;","S"):replace("&#199;","C"):replace("&#220;","U"):
replace("&#214;","O"):replace("&#305;","i"):replace("&#287;","g"):replace("&#351;","s"):replace("&#231;","c"):
replace("&#252;","u"):replace("&#246;","o")}","${emlak2.6:replaceAll("<|>",""):replace(" im²/i",""):replace(" TL",""):replace(".","")}")
```

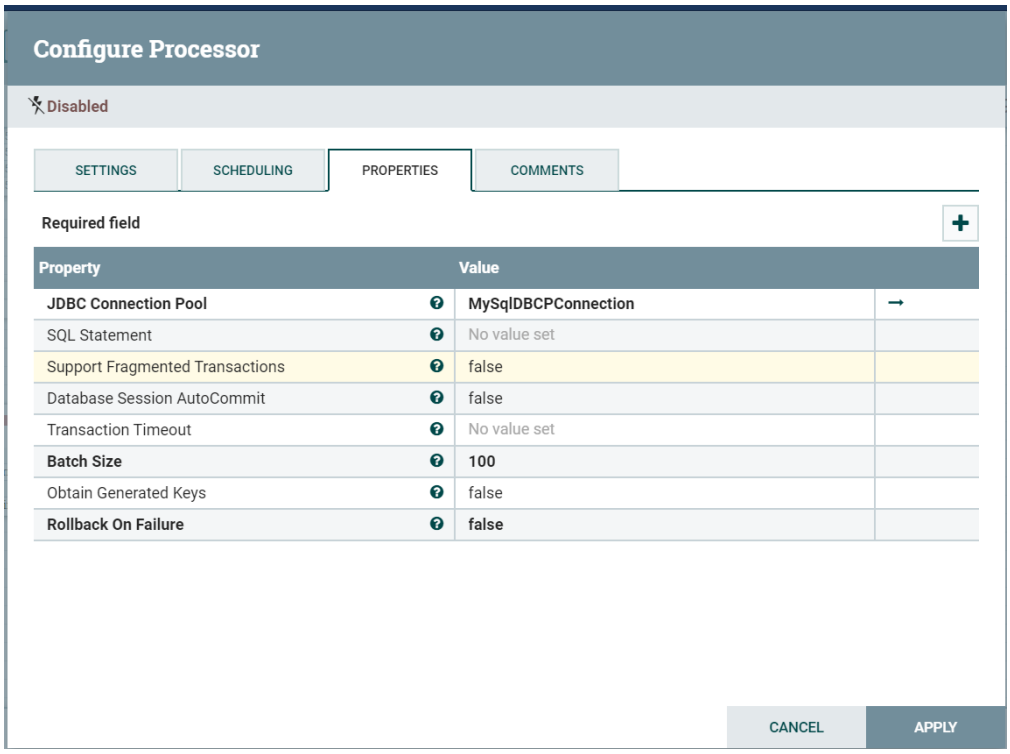
After this, code block successfully transformed to insert scripts.

```
INSERT INTO bitirme.hurriyetemlak_v2 (provience, district, neighborhood, publish_date, room, square, price)
VALUES ("İstanbul","Adalar","Buyukada-Maden","01.12.2019","5+2","210","830000")
```

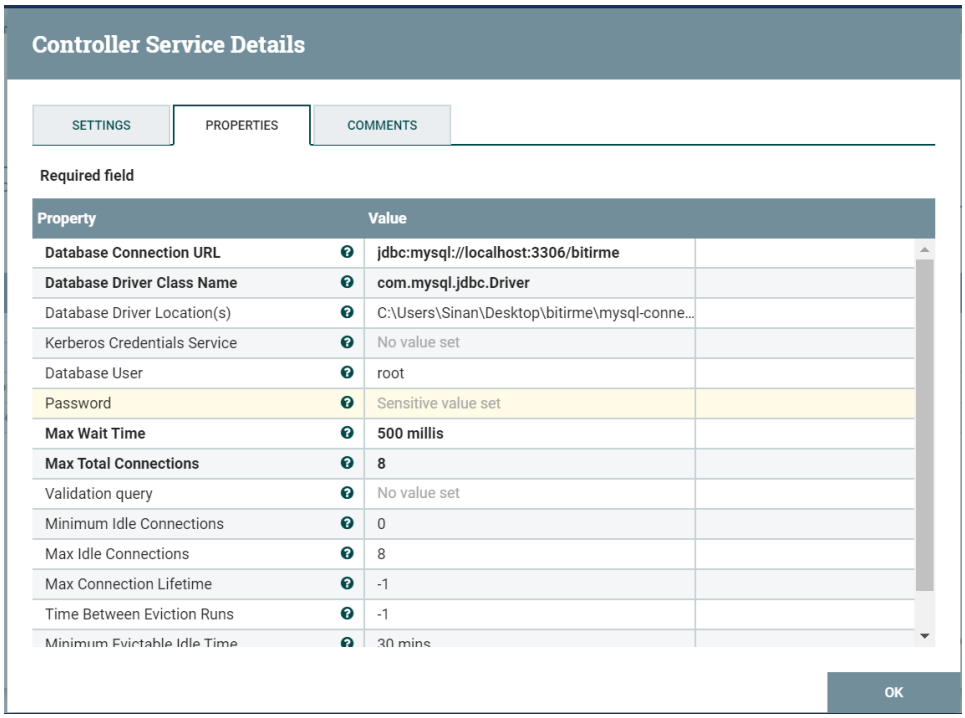
Before sending insert script to MySql database, a table which satisfies this attributes created on ‘bitirme’ database with ‘sinan’ user.



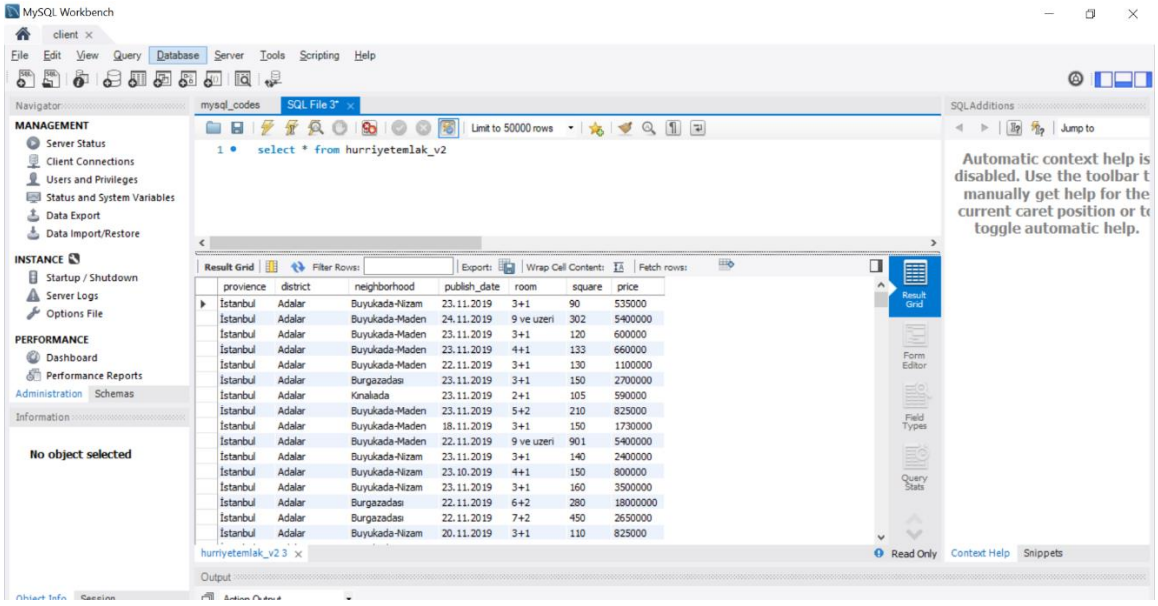
After creating ‘hurriyetemlak_v2’ table, last processor of the data flow, ‘PutSQL’ processor sends sql insert scripts to MySql database. The processor setting is:



To get processor work, connection properties of MySql database has to be defined in ‘JDBC connection pool’ controller service.



The design explained above has worked for 12,5 hours. It has download 60202 records from Hürriyetemlak.com.



After this process, another table created from hurriyetemlak_v2 to prepare data analyses which would have been made in Pycharm.

```sql
create table hurriyetemlak_v4 as
select lower(provience) as provience,lower(district) as district,lower(neighborhood) as neighborhood,
convert(price, signed int) as price,convert(square, signed int) as square,
convert(case
when room ='3+1' then 4
when room ='9 ve uzeri' then 10
when room ='4+1' then 5
when room ='2+1' then 3
when room ='5+2' then 7
when room ='6+2' then 8
when room ='7+2' then 9
when room ='5+1' then 6
when room ='6+1' then 7
when room ='7+4 ve uzeri' then 12
when room ='3+2' then 5
when room ='4+2' then 6
when room ='8+2' then 10
when room ='8+3' then 11
when room ='5+3' then 8
when room ='7+1' then 8
when room ='7+3' then 10
when room ='1+1' then 2
when room ='6+3' then 9
when room ='8+1' then 9
when room ='8+4 ve uzeri' then 13
when room ='6+4 ve uzeri' then 11
when room ='2+2 ve uzeri' then 4
when room ='4+4 ve uzeri' then 8
when room ='4+3' then 7
when room ='0' then 0
when room ='5+4 ve uzeri' then 10
when room ='2' then 2
when room ='3+3 ve uzeri' then 6
when room ='1+0' then 1
when room ='3' then 3
when room ='4' then 4
when room ='5' then 5
when room ='7' then 7
when room ='6' then 6
when room ='8' then 8
else room
end, signed int) as room
from hurriyetemlak_v2;
```