

**T.C**

**ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**İLERİ VERİ TABANI DERSİ**

**PROJE KONUSU**

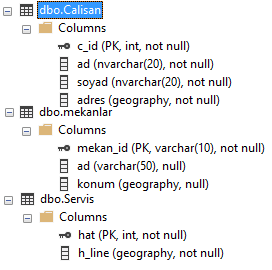
Spatial Type’lar ile Web Uygulaması

**NESLİHAN AYDIN**

**140401013**

Bir şirketin çalışanlarının kullanması gereken servisleri gösteren projede, kişi çalışan numarasını girip, öğrenme tuşuna bastığında, ekranda çalışanın evi Point ,kullanması gereken servisin geçtiği güzergahi Multiline , şirketi de Polygon şeklinde haritada gösterir.

Veritabanı yapısı aşağıdaki şekildedir.



Çalışan eklerken kullanılan sorgular :

insert into Calisan VALUES (

137333345,

'Sevim',

'Arslan',

geography::STGeomFromText('POINT(40.111068 26.404773)', 4326)

);

insert into Calisan VALUES (

1234567,

'Elmas',

'Turpcu',

geography::STGeomFromText('POINT(40.131213 26.437002)', 4326)

);

insert into Calisan VALUES (

199333345,

'Neslihan',

'Aydin',

geography::STGeomFromText('POINT(40.111068 26.403442)', 4326)

);

insert into Calisan VALUES (

999123345,

'Zeynep',

'Turker',

geography::STGeomFromText('POINT(40.142168 26.437002)', 4326)

);

insert into Calisan VALUES (

8883345,

'Fatma',

'Aydın',

geography::STGeomFromText('POINT(40.131322 26.437552)', 4326)

);

Sadece iki servis olduğu için servis eklerken iki adet sorgu kullanılmıştır:

insert into Servis VALUES (

2,

geography::STMLineFromText('MULTILINESTRING((40.115598 26.414472, 40.116514 26.412412, 40.113758 26.413099, 40.106534 26.402605, 40.117594 26.41063, 40.125436 26.409686, 40.133015 26.408913, 40.14184 26.407926, 40.145874 26.407926, 40.153582 26.414106, 40.150448 26.42503))', 4326)

);

insert into Servis VALUES (

1,

geography::STMLineFromText('MULTILINESTRING ((40.115598 26.414472,40.116514 26.412412,40.113758 26.413099,40.106534 26.402605,40.117594 26.410630,40.125436 26.409686,40.133015 26.408913,40.141840 26.407926,40.145874 26.407926,40.153582 26.414106,40.150448 26.425030))', 4326)

);

Projedeki mekanları eklemek için şu sorgular yazılmıştır :

insert into mekanlar VALUES (

1,

'Sirket',

geography::STGeomFromText('POLYGON((40.113385 26.416874 , 40.111907 26.419320, 40.110134 26.414471, 40.112168 26.413441,40.113385 26.416874 ))', 4326)

);

insert into mekanlar VALUES (

2,

'17 Burda',

geography::STGeomFromText('POLYGON((40.122365 26.410320 , 40.122298 26.413109, 40.125677 26.410577, 40.125709 26.409976,40.122365 26.410320 ))', 4326)

);

insert into mekanlar VALUES (

3,

'Universite',

geography::STGeomFromText('POLYGON((40.113385 26.416874 , 40.111907 26.419320, 40.110134 26.414471, 40.112168 26.413441,40.113385 26.416874 ))', 4326)

);

insert into mekanlar VALUES (

4,

'Troypark',

geography::STGeomFromText('POLYGON((40.127433 26.411363 , 40.126710 26.412264, 40.125626 26.410891, 40.126741 26.410076,40.127433 26.411363 ))', 4326)

);

insert into mekanlar VALUES (

5,

'KYK',

geography::STGeomFromText('POLYGON((40.132679 26.411235 , 40.132941 26.410334, 40.133465 26.411021, 40.133366 26.411236,40.132679 26.411235 ))', 4326)

);

Projede kullanıcı çalışan numarasını girip gönder tuşuna bastığına aşağıdaki kod parçaları çalışır;

SqlCommand da = new SqlCommand("select distinct Calisan.c\_id, Calisan.ad, Servis.hat from Calisan, Servis where Servis.h\_line.STDistance(Calisan.adres) < 950 and Calisan.c\_id="+TextBox1.Text, baglan);

SqlDataReader reader = da.ExecuteReader();

try

{

while (reader.Read())

{

TextBox1.Text = reader[0].ToString();

TextBox2.Text = reader[1].ToString();

TextBox3.Text = reader[2].ToString();

}

}

finally

{

reader.Close();

}

Bu sorgu ile ekranda kullanıcının adı ve kullanması gereken hattın numarası yazar.

Ayrıca gönder tuşuna basıldığı zaman veritabanından diğer veri çekme işlemleri de aynı zamanda yapılır.

Çalışanın adresini haritada point olarak göstermek için, veritabanından konumunu çektikten sonra STAsText( ) kullanılarak bir TextBox’a çıktısı yazdırılır.Sonrasında parcala adındaki fonksiyona parametre olarak Textbox’ın içindeki Text gönderilir.

baglan.Open();

SqlCommand da21 = new SqlCommand("select (Calisan.adres).STAsText() from Calisan where Calisan.c\_id=" + TextBox1.Text, baglan);

SqlDataReader reader21 = da21.ExecuteReader();

try

{

while (reader21.Read())

{

TextBox41.Text = reader21[0].ToString();

parcala(TextBox41.Text);

}

}

finally

{

reader21.Close();

}

baglan.Close();

parcala fonksiyonun içeriği şu şekildedir ;

public void parcala(string text)

{

string lat = "";

string lng = "";

int durum = 0;

string temp = "";

for (int i = 0; i < text.Length; i++)

{

if(text[i] == '(')

{

durum = 1;

i++;

}

if (durum==1)

{

if(text[i]==' ')

{

lat = temp;

i++;

temp = "";

}

if (text[i]==')')

{

lng = temp;

}

temp += text[i];

}

}

Label4.Text = lat;

Label5.Text = lng;

}

parcala fonksiyonu ile çalışanın evinin konumu lat ve lng olarak ayrılır. Sonrasında script kodlarının içerisinde getElementById kullanılarak lat ve lng değerleri çekilir.

var latt = document.getElementById("Label4");

var lngg = document.getElementById("Label5");

if (latt.value != "")

{

var pos = { lat: parseFloat(latt.textContent), lng: parseFloat(lngg.textContent) };

var marker = new google.maps.Marker({

position: pos,

map: map

});

}

marker.setMap(map);

Diğer yandan şirketin konumunu veritabanından çekmek için şu kodlar çalıştırılır ;

baglan.Open();

SqlCommand da2 = new SqlCommand("select (mekanlar.konum).STAsText() from mekanlar where mekanlar.mekan\_id = 1 ", baglan);

SqlDataReader reader2 = da2.ExecuteReader();

try

{

while(reader2.Read())

{

TextBox4.Text = reader2[0].ToString();

}

}

finally

{

reader2.Close();

}

baglan.Close();

Şirketin konumunu çekilip, polygon metninin içeriği textbox a yazıldıktan sonra script kodları içerisinde parçalama işlemi yapılır.

var polycoord = document.getElementById("TextBox4");

if (polycoord.value != "") {

var plats = new Array(); var plngs = new Array();

var ptext = polycoord.value; var plat = ""; var plng = ""; var pdurum = 0; var ptemp = ""; var pi = 0;

for (pi = 0; pi < ptext.length; pi++)

{ if (ptext[pi] == '(') {

pdurum = 1;

pi = pi + 2; // POLYGON ((40.113385 26.416874, 40.111907 26.41932, 40.110134 26.414471, 40.112168 26.413441, 40.113385 26.416874))

}

if (pdurum == 1) {

if (ptext[pi] == ' ') {

plat = ptemp;

plats.push(plat);

pi++;

ptemp = "";

}

if (ptext[pi] == ',') {

plng = ptemp;

plngs.push(plng);

pi = pi + 2;

ptemp = "";

}

if (ptext[pi] == ')') {

plng = ptemp;

plngs.push(plng);

}

ptemp += ptext[pi];

}

}

var SirketPoly = [

{ lat: parseFloat(plats[0]), lng: parseFloat(plngs[0]) },

{ lat: parseFloat(plats[1]), lng: parseFloat(plngs[1]) },

{ lat: parseFloat(plats[2]), lng: parseFloat(plngs[2]) },

{ lat: parseFloat(plats[3]), lng: parseFloat(plngs[3]) },

{ lat: parseFloat(plats[4]), lng: parseFloat(plngs[4]) }

];

// Construct the polygon.

var bermudaTriangle = new google.maps.Polygon({

paths: SirketPoly,

strokeColor: '#FF0000',

strokeOpacity: 0.8,

strokeWeight: 2,

fillColor: '#FF0000',

fillOpacity: 0.35

});

bermudaTriangle.setMap(map);

}

Çalışanın kullanması gereken hattı güzergahı da şu şekilde veritabanından çekilir;

baglan.Open();

SqlCommand da3 = new SqlCommand("select (Servis.h\_line).STAsText() from Servis where Servis.hat = " + TextBox3.Text, baglan);

SqlDataReader reader3 = da3.ExecuteReader();

try

{

while (reader3.Read())

{

TextBox42.Text = reader3[0].ToString();

}

}

finally

{

reader3.Close();

}

baglan.Close();

}

Ardından çekilen multiline metni script kodları içerisinde parçalanır. Kullanılan script kodu aşağıdaki gibidir.

var linecoord = document.getElementById("TextBox42");

if (linecoord.value != "")

{

var llats = new Array();

var llngs = new Array();

var ltext = linecoord.value;

var llat = "";

var llng = "";

var ldurum = 0;

var ltemp = "";

var li = 0;

for (li = 0; li < ltext.length; li++)

{

if (ltext[li] == '(')

{

ldurum = 1;

li = li + 2;

}

if (ldurum == 1)

{

if (ltext[li] == ' ') {

llat = ltemp;

llats.push(llat);

li++;

ltemp = "";

}

if (ltext[li] == ',') {

llng = ltemp;

llngs.push(llng);

li = li + 2;

ltemp = "";

}

if (ltext[li] == ')') {

llng = ltemp;

llngs.push(llng);

}

ltemp += ltext[li];

}

}

var ServisLine = [

{ lat: parseFloat(llats[0]), lng: parseFloat(llngs[0]) },

{ lat: parseFloat(llats[1]), lng: parseFloat(llngs[1]) },

{ lat: parseFloat(llats[2]), lng: parseFloat(llngs[2]) },

{ lat: parseFloat(llats[3]), lng: parseFloat(llngs[3]) },

{ lat: parseFloat(llats[4]), lng: parseFloat(llngs[4]) },

{ lat: parseFloat(llats[5]), lng: parseFloat(llngs[5]) },

{ lat: parseFloat(llats[6]), lng: parseFloat(llngs[6]) },

{ lat: parseFloat(llats[7]), lng: parseFloat(llngs[7]) }

];

var otobusler = new google.maps.Polyline({

path: ServisLine,

geodesic: true,

strokeColor: '#FF0000',

strokeOpacity: 1.0,

strokeWeight: 2

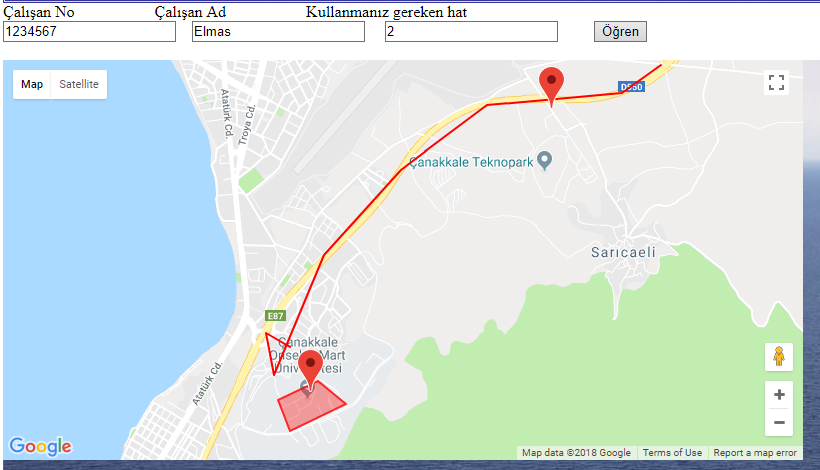
});

otobusler.setMap(map);

}

Çalışanın kullanması gereken hattın güzergahı da haritaya eklemiş olur.

Programın ekran görüntüsü aşağıdaki şekildedir.



Raporun bulunduğu blog : <https://goo.gl/1To18d>

Projenin kaynak kodları : <https://github.com/neslihanaydin/Ileri-veri-Tabani>