

Hafta 10 Ders Föyü

Veri Tabanında PL/pgSQL Program Kodu Yazmak

1) Harita üzerindeki toplam ada sayısını veren programı yazın.

```
1 do
2 $$
3 declare
4     ada_sayisi integer;
5 begin
6     select count(*) into ada_sayisi from ada;
7     raise notice 'ada sayısı=%',ada_sayisi;
8
9 end$$;
```

2) yolların adları ve yol uzunluklarını listeleyen program yazınız.



Sorgunun sonucunda birden fazla kayıt dönecektir. Birden fazla yol var ve her yolun uzunluğu isteniyor. Bu işlem normal bir SQL sonucunda geriye değer döndürecektir ve sonuçlar listelenecektir. Fakat işlem bu sefer bir programlama dili ile yapılıyor. Bir den fazla kayıttan sonucunun programlama dili ile listelenmesi veya işleme koyulması için bir döngü işlemi (for loop döngüsü) kullanılmalı.

Şekil 1 4. Satırda record veri tipinde bir değişken tanımlanıyor. record değişkeni normal değişken tiplerinden farklı olarak geri dönen sonucu tutmak için kullanılır.

Şekil 1 6. Satır ile 8. Satır arasında bir for loop döngüsü kullanımı var. Altta yazan SQL sorgu cümlesi sonucu birden fazla yol objesinin hem ad bilgisi hem de uzunluk bilgisi gelecek. Uzunluk sonuçları *uzunluk* adlı değişkene aktarılıyor.

```
select ad, st_length(geom_yol) as uzunluk
from yol
```

Aşağıdaki satır sayesinde sorgunun her bir sonucu sırasıyla sonuç adlı record veri tipindeki değişkene aktarılıyor.

```
sonuc in (select ad, st_length(geom_yol) as uzunluk from yol)
```

Aşağıdaki satır ile ekrana çıktı olarak listeleniyor.

```
raise notice 'yolun adı = % yolun uzunluğu = %',sonuc.ad,sonuc.uzunluk;
```

For loop – End Loop aralığı ile liste sonuna kadar işlem devam ediyor.

```
1 do
2 $$
3 declare
4     sonuc record;
5 begin
6     for sonuc in (select ad,st_length(geom_yol) as uzunluk from yol) loop
7         raise notice 'yolun adı=% yolun uzunluğu=%',sonuc.ad,sonuc.uzunluk;
8     end loop;
9 end $$;
```

Şekil 1

3) 1433 numaralı ada içindeki 10 numaralı parsel 50 metre yakındaki yolun adını veren program kodunu yazın.

```
1 do
2 $$
3 declare
4     yoladi yol.ad%type;
5 begin
6     select yol.ad into yoladi from yol,parsel,ada where ada.ada_no=1433 and parsel.parsel_no=10
7     and st_contains(ada.geom_ada,parsel.geom_parsel)
8     and st_dwithin(yol.geom_yol,parsel.geom_parsel,50.000);
9     raise notice 'yolun adı=%',yoladi;
10 end$$;
```

```
NOTICE: yolun adı=1014. sokak
DO
```

```
Query returned successfully in 42 msec.
```

3.1) (3. Soru değiştiriliyor) 1433 numaralı ada içindeki 10 numaralı parsel en yakın yolun adını veren program kodunu yazınız.

Cevap: soruda istenilen parsel en yakın yolun adı. Bu işlem parselin cephesi olan yolu bulmak için de kullanılabilir.

```

1 do
2 $$
3 declare
4     yoladi yol.ad%type;
5 begin
6     select yol.ad into yoladi
7     from yol,parsel,ada
8     where ada.ada_no=1433
9     and parsel.parsel_no=10
10    and st_within(parsel.geom_parsel,ada.geom_ada)
11    order by parsel.geom_parsel <-> yol.geom_yol ASC
12    limit 1;
13    raise notice 'yolun adı=%',yoladi;
14 end$$;

```

4) 3.1 numaralı soruyu hafta 6 içinde hazır bir fonksiyon haline dönüştürülmesi isteniyor. Bu sayede istendiğinde ada ve parsel numarası verilen parsele yakın olan yolun adı öğrenilmesi sağlanır.

```

1 create or replace function parselecephoyoladi(adanodeg integer,parselnodeg integer) returns text as
2 $$
3 declare
4     yoladi yol.ad%type;
5 begin
6     select yol.ad into yoladi
7     from yol,parsel,ada
8     where ada.ada_no=adanodeg
9     and parsel.parsel_no=parselnodeg
10    and st_within(parsel.geom_parsel,ada.geom_ada)
11    order by parsel.geom_parsel <-> yol.geom_yol ASC
12    limit 1;
13 return yoladi;
14 end$$ Language plpgsql;

```

CREATE FUNCTION

Query returned successfully in 28 msec.

```
overlaps_nd(gidx, geometry)
overlaps_nd(gidx, gidx)
parselecepheyladi(adanodeg integer, parselnodeg integer)
path(geometry)
pgis_asflatgeobuf_finalfn(internal)
pgis_asflatgeobuf_transfn(internal, anyelement)
```

```
1 select parselecepheyladi(1433, 10);
```

	parselecepheyladi 
1	1014. sokak

5) 1001 numaralı enerji nakil hattının üzerinden geçip güzergâhına devam ettiği parsel numaraları, adaların numaralarını verecek fonksiyonu yazınız.

Cevap:

Fonksiyon Şekil 2’de olduğu gibi oluşturulması gerekli. Hazırlanan fonksiyon 3. Sorudakinden farklı yazıldı. Şekil 2 2. Satıra dikkat edilirse, returns kelimesinden sonra bir tablo sonucu geriye dönecek. 3. Soruda geriye bir değer dönüyorken bu soruda birden fazla kaydın olduğu bir tablo geriye dönecek. Bu işlemde önemli olan kriterler:

returns table(parametre veritipi) → parametrenin veri tipi tablo kullanılacak tablonun veri sorguda kullanılan tablodaki karşılık gelen sahanın veri tipi ile aynı olmalıdır,

7. satırdaki sorgu cümlesinde saha adları sıralaması ile 2. Satırdaki table() içindeki karşılık gelen saha sıralamaları aynı olmalıdır.

6. satırdaki return QUERY kelimesi sabittir. Anlamı “sonraki satırda yazılacak sorgu cümlesi sonucunu tabloya aktaracağım.” karşılık gelmektedir.

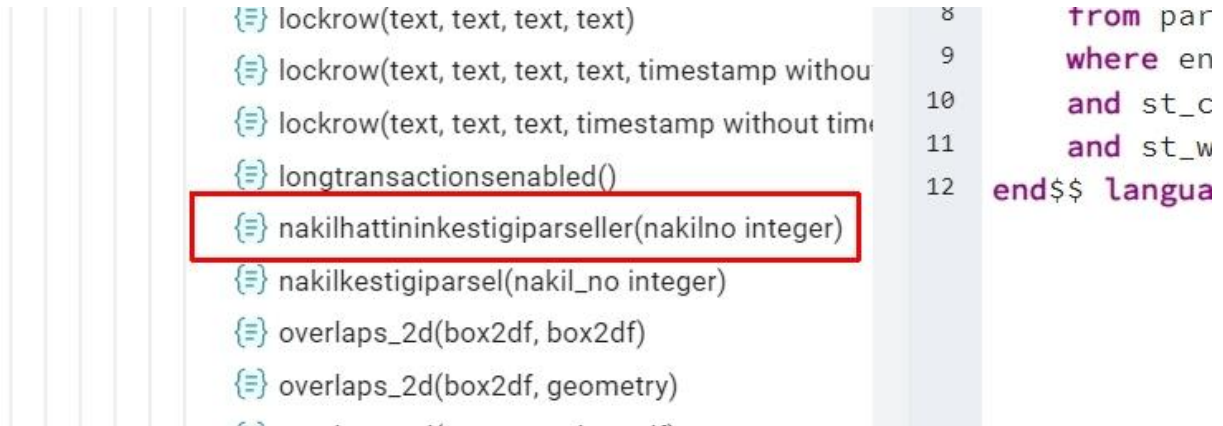
```

1 create or replace function nakilhattinkestigiparseller(nakilno integer)
2 returns table(adaninnumarasi integer, parselnumarasi integer,parselkimlik double precision,parselgeo geometry)
3 as
4 $$
5 begin
6     return QUERY
7     select ada.ada_no,parsel.parsel_no, parsel.parsel_id,parsel.geom_parsel
8     from parsel,ada,enerji_hatti
9     where enerji_hatti.nakil_no=nakilno
10    and st_crosses(enerji_hatti.geom_enerji,parsel.geom_parsel)
11    and st_within(parsel.geom_parsel,ada.geom_ada);
12 end$$ language plpgsql;

```

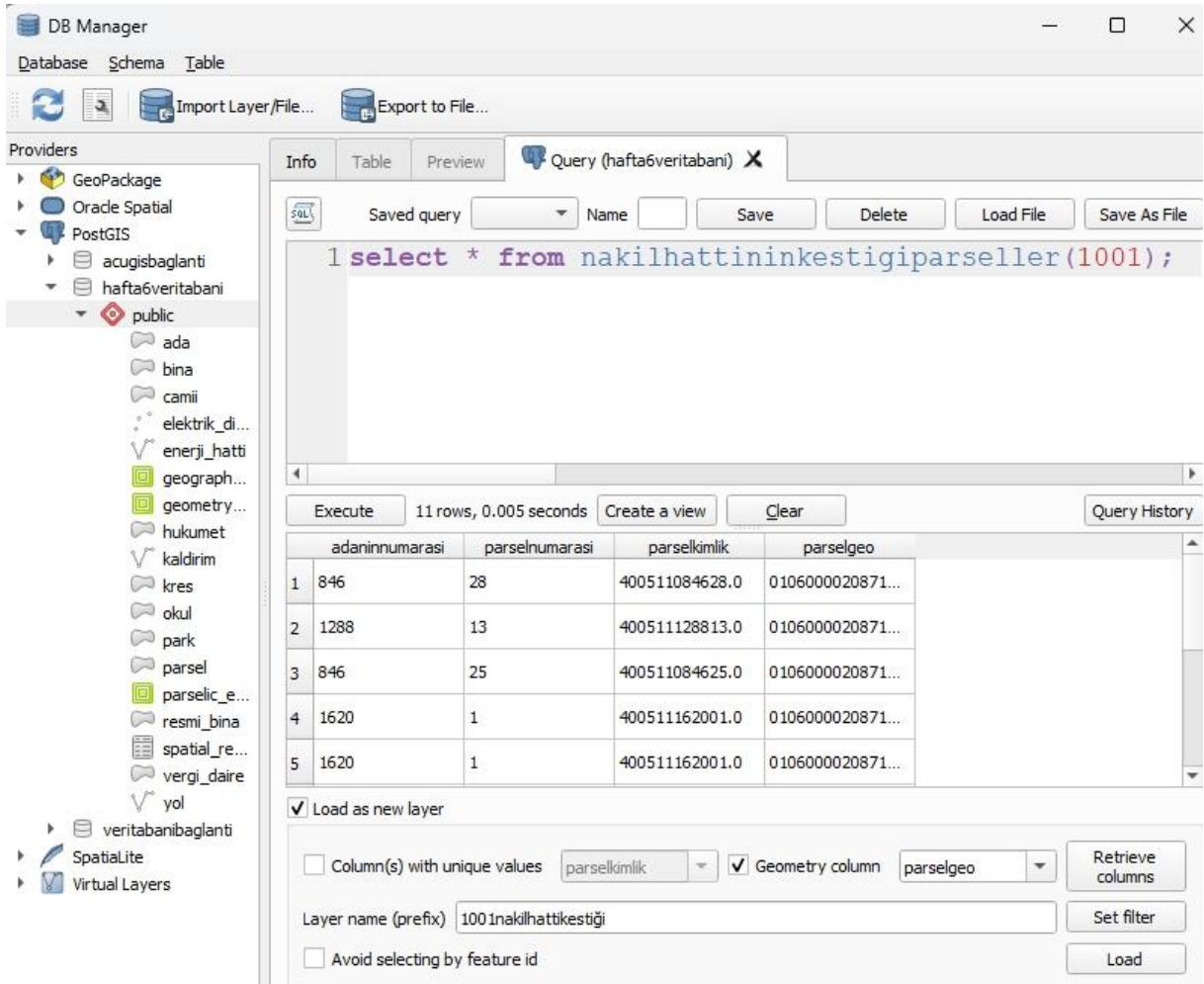
Şekil 2

Fonksiyon çalıştırdıktan sonra veri tabanı içindeki functions kısmındaki fonksiyonlar yenilenmeli ve fonksiyon adı kontrol edilmelidir (Şekil 3).



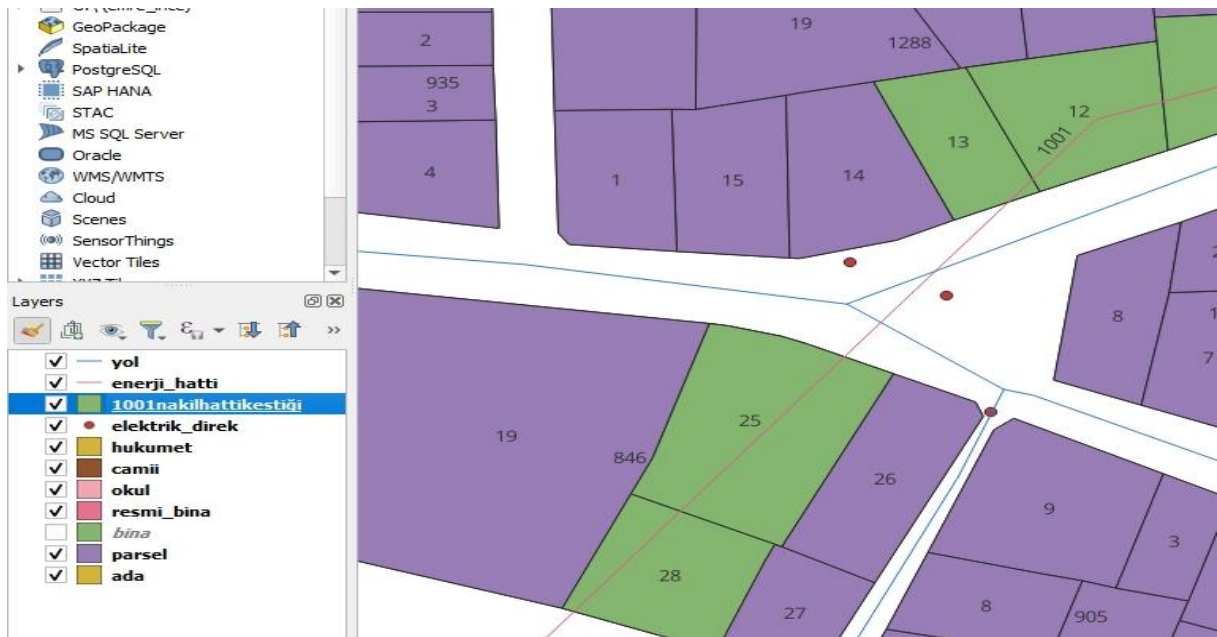
Şekil 3

Fonksiyonun qgis yazılımında çalıştırılması işlemi için, Qgis yazılımında PostgreSQL içinde hazırlanmış olan veri tabanına bağlantı kurulması gereklidir. Sonrasında *Database (Veritabanı)* menüsü altında *DB Manager (Veritabanı Yönetimi)* penceresinde sorgu çalıştırılabilir (Şekil 4).



Şekil 4

Şekil 5 sonucun qgis penceresinde görünümü.



Şekil 5

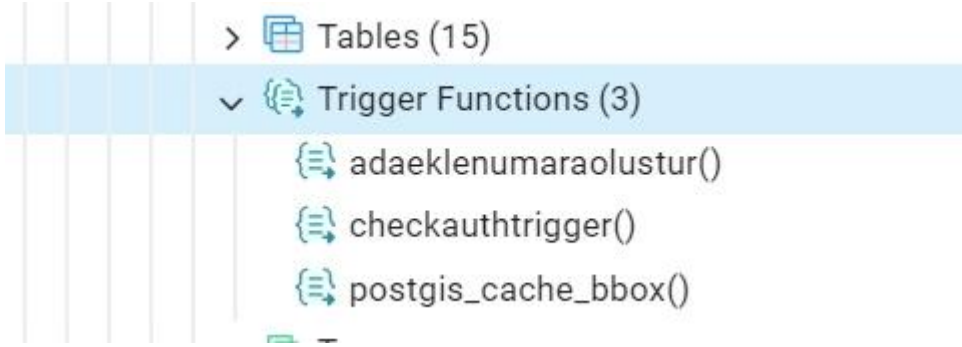
6) ada tabakasına yeni bir ada çizilip kaydedildiğinde, yeni oluşturulan ada objesinin ada numarası bilgisi otomatik doldurulması istenmektedir. Yeni ada objesi için ada numarası, ada tabakasındaki ada objeleri içinde en büyük ada numarasına sahip olan tespit edilip, bu numaradan bir fazlası olacak şekilde oluşturulmalıdır. Ada numarası oluşturulduktan sonra ada objesinin kimlik numarası da otomatik oluşturulması istenmektedir. Bu işlemlerin yapılması için bir fonksiyon oluşturulması istenmektedir.

Cevap:

İstenilen işlemi yapacak fonksiyonun otomatik çalışması gerekli. Fonksiyon yeni bir ada objesi oluştuğunda fonksiyon çalışmalı. Ada objesi eklendiği anda (eklenmesi sonrasında) fonksiyon hemen çalışmalı. Bu tür işlem öncesi (örneğin obje ekleme işlemi öncesi) veya işlem sonrası (örneğin ekleme işlemi sonrası) fonksiyon çalışması için bir tetikleyici konulmalı. Tetikleyici işlemi fark eder ve gerekli fonksiyonun çalışmasını sağlar. Fonksiyon da tetiklenecek olarak oluşturulmalıdır. Şekil 6 tetiklenen fonksiyon oluşturma PL/pgSQL kodudur.

```
1 CREATE OR REPLACE FUNCTION public.adaeklenumaraolustur() RETURNS trigger as
2 $$
3 declare
4     adanoenyuksek integer;
5     kayitsayi integer;
6     adakimlikyeni double precision;
7 begin
8     --toplam kayıt sayısını soruyorum
9     select count(*) into kayitsayi from ada;
10    --eğer kayıt sayısı yeni kayıt ile 1 ise
11    if kayitsayi=1 then
12        NEW.ada_no=101;
13        NEW.ada_id=400511101;
14    elsif kayitsayi>1 then
15        select max(ada_no) into adanoenyuksek from ada;
16        if adanoenyuksek>=101 and adanoenyuksek<999 then
17            NEW.ada_no := adanoenyuksek + 1;
18            NEW.ada_id := 400511 * 1000::double precision + (adanoenyuksek + 1);
19        elsif adanoenyuksek>=999 then
20            NEW.ada_no := adanoenyuksek + 1;
21            NEW.ada_id := 400511 * 10000::double precision + (adanoenyuksek + 1);
22        end if;
23    end if;
24    return NEW;
25 end
26 $$ language plpgsql;
```

Şekil 6



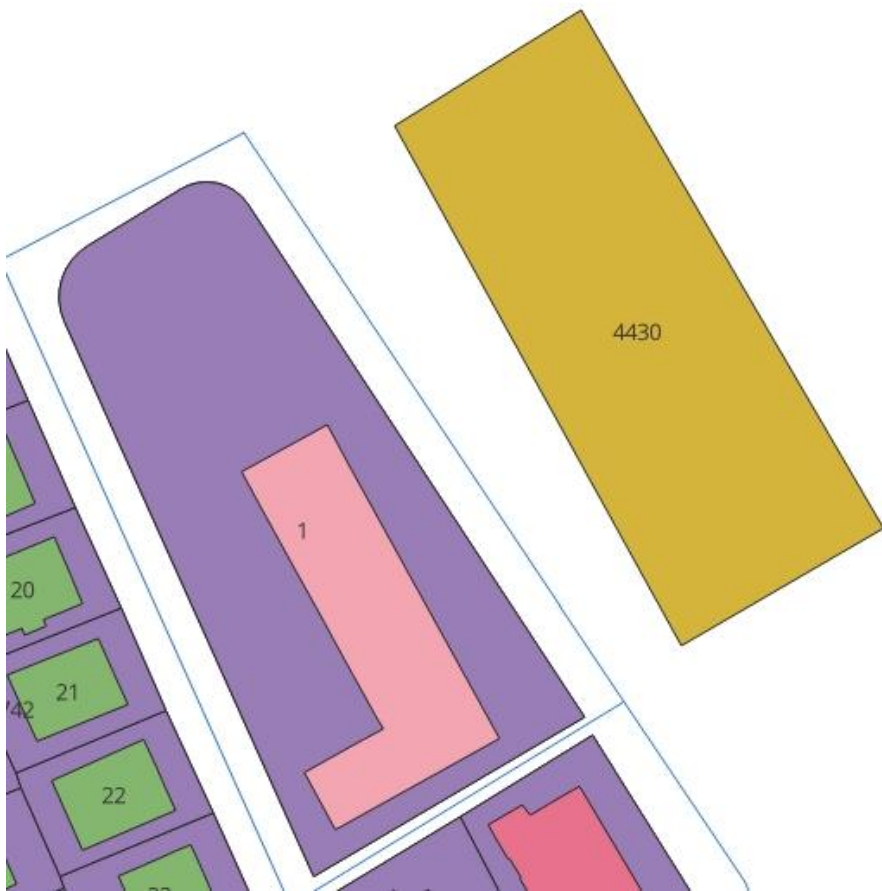
Şekil 7 adaeklenumaraolustur() fonksiyonunun tetikleyici PL/pgSQL kodudur.

```
1 CREATE TRIGGER yeniadaekletrigger
2 BEFORE INSERT ON ada
3 FOR EACH ROW
4 EXECUTE FUNCTION adaeklenumaraolustur();
```





Şekil 7



Çizim yapıp düzenlemeyi kaydet tuşuna basıldığı anda işlemlerin yapılması isteniyor.



/

	 objectid bigint 	ada_no integer 	ada_id double precision 
	[null]	4430	4005114430
	8	4429	4005114429
	9	4428	4005114428
	1	3943	4005113943

HAFTA 10 ÖDEVİ

6. soruda çizimini Qgis coğrafi bilgi sistemleri yazılımında yaptığınız 4430 numaralı ada içinde yeni parseller çizilecek. Parsel çizerken nokta yakalama modu açık olacak ve 4430 numaralı ada sınırları yakalama modu açıkken dikkate alınarak parseller çizilecek. Sizden istenen çizim işi bittikten sonra kayıt işlemi **yapıldığında** Parsel numaraları otomatik olarak oluşması parsel tabakasında ki parsel_no sahasına kaydedilmesi, parselin içinde kaldığı ada kimliği (ada_id) otomatik olarak parsel tabakasındaki ada_id sahasına kayıt altına alınması, parsel tabakasındaki parsel_id sahasındaki veri otomatik olarak sahaya kayıt altına alınması. Bu işlemleri yapan fonksiyon ve fonksiyon tetikleyicisinin yazılması istenmektedir.