

# Bartosz Romaniuk

Prosimy o:

- zainstalowanie bazy danych PostgreSQL z rozszerzeniem przestrzennym (POST GIS)
- w bazie danych należy stworzyć tabelę punkty (X,Y,Z)
- w języku C# lub C++ napisać program który połączy się z bazą danych
- proszę napisać funkcję która wygeneruje losowe punkty w obszarze granic Polski i zapisze je w tabeli punkty (min 1000 punktów, dla wartości Z czyli wysokości proszę przyjąć zakres od 0 do 300 metrów)
- proszę napisać test który będzie sprawdzał czy generowane punkty są w odległości min 3 km od siebie
- proszę stworzyć tabelę województwa i zaimportować tam granice województw
- proszę napisać funkcję która dla każdego województwa wypisze wszystkie wygenerowane punkty które zawierają się w jego obrysie

## **- zainstalowanie bazy danych PostgreSQL z rozszerzeniem przestrzennym (POST GIS)**

Ze strony <https://www.postgresql.org/download/windows/> pobrałem najnowszą wersję PostgreSQL na platformę Windows. Podczas instalacji wybrałem opcję do zainstalowania rozszerzenia przestrzennego Post Gis.

## **- w bazie danych należy stworzyć tabelę punkty (X,Y,Z)**

Skrypt do utworzenia tabeli:

```
CREATE TABLE public.punkty
(
    id integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1
MAXVALUE 2147483647 CACHE 1 ),
    x text COLLATE pg_catalog."default",
    y text COLLATE pg_catalog."default",
    z text COLLATE pg_catalog."default",
    punkt_txt text COLLATE pg_catalog."default",
    punkt_geom geometry,
    CONSTRAINT punkty_pkey PRIMARY KEY (id)
)

TABLESPACE pg_default;

ALTER TABLE public.punkty
    OWNER to postgres;
```

**- w języku C# lub C++ napisać program który połączy się z bazą danych**

Napisałem program w języku C#. Kod źródłowy znajduje się w pliku „ Form1.cs „  
W celu sprawdzenia czy napisany program działa poprawnie do aplikacji okienkowej został dodany przycisk , który tworzy bazę danych w PostgreSQL. Do poprawnego połączenia został dodany NuGet do PostgreSQL „Npgsql”.

**- proszę napisać funkcję która wygeneruje losowe punkty w obszarze granic Polski i zapisze je w tabeli punkty (min 1000 punktów, dla wartości Z czyli wysokości proszę przyjąć zakres od 0 do 300 metrów)**

Za pomocą PostGis ShapeFile Import/Export Manager został zaimportowany plik „panstwo.shp” tworząc jednocześnie tabele „panstwo” w której znajdują się wszystkie niezbędne dane.

Skrypt SQL do funkcji generujące losowe punkty w obszarze granic Polski:

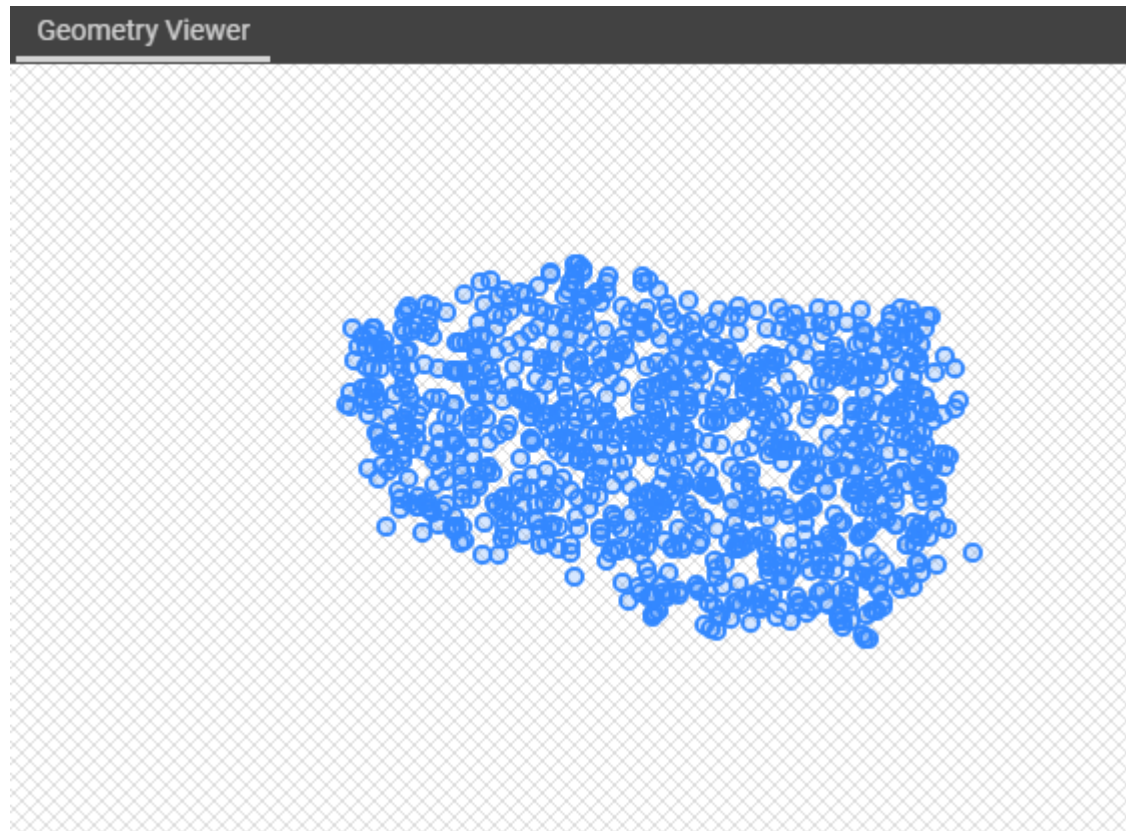
```
CREATE OR REPLACE FUNCTION public.randompointsinpolygon(  
    geom geometry,  
    num_points integer)  
RETURNS SETOF geometry  
LANGUAGE 'plpgsql'  
  
COST 100  
VOLATILE  
ROWS 1000  
  
AS $BODY$DECLARE  
    target_proportion numeric;  
    n_ret integer := 0;  
    loops integer := 0;  
    x_min float8;  
    y_min float8;  
    x_max float8;  
    y_max float8;  
    srid integer;  
    punkt geometry;  
BEGIN  
  
    SELECT ST_XMin(geom), ST_YMin(geom), ST_XMax(geom), ST_YMax(geom), ST_SRID(geom)  
    INTO x_min, y_min, x_max, y_max, srid;  
  
    SELECT ST_Area(geom)/ST_Area(ST_Envelope(geom)) INTO target_proportion;  
    WHILE n_ret < num_points LOOP  
        loops := loops + 1;  
        SELECT ST_SetSRID(ST_MakePoint(random()*(x_max - x_min) + x_min,  
            random()*(y_max - y_min) + y_min),  
            srid) INTO punkt;  
        IF ST_Contains(geom, punkt) THEN  
            n_ret := n_ret + 1;  
            RETURN NEXT punkt;  
        END IF;  
    END LOOP;  
END$BODY$;  
  
ALTER FUNCTION public.randompointsinpolygon(geometry, integer)  
OWNER TO postgres;
```

Po napisaniu funkcji użyłem zapytania do wygenerowania 1000 punktów i zapisania go tabeli „punkty”. :

```
Insert into punkty (x,y,punkt_txt,punkt_geom)
Select ST_X(ST_AsText(ST_GeometryN(RandomPointsInPolygon((ST_Dump(ST_SetSRID(geom,
2180))))).geom, 1000),1)),
ST_Y(ST_AsText(ST_GeometryN(RandomPointsInPolygon((ST_Dump(ST_SetSRID(geom,
2180))))).geom, 1000),1)),
ST_AsText(ST_GeometryN(RandomPointsInPolygon((ST_Dump(ST_SetSRID(geom, 2180))))).geom,
1000),1)),
RandomPointsInPolygon((ST_Dump(ST_SetSRID(geom, 2180))))).geom, 1000) from panstwo;
```

Wyjście Danych						
Plan zapytania						
Komunikaty						
Notifications						
Geometry Viewer						
	id [PK] integer	x text	y text	punkt_txt text	punkt_geom geometry	
1		1 19.0818855518555	52.6661231965112	POINT(19.0818855518555 52.6661231965112)	010100002084080000013C9773F614_	
2		2 16.4033162856778	51.7970740761781	POINT(16.4033162856778 51.7970740761781)	01010000208408000028EE70BC3F67_	
3		3 17.3899239848464	51.7731096880282	POINT(17.3899239848464 51.7731096880282)	0101000020840800005DD7EA0ED263_	
4		4 21.6501514838865	53.4791543613702	POINT(21.6501514838865 53.4791543613702)	010100002084080000FBBCE05370A6_	
5		5 18.1649831647047	53.3246530920745	POINT(18.1649831647047 53.3246530920745)	01010000208408000027CC30563C2A_	
6		6 21.1806555668057	53.9346836937005	POINT(21.1806555668057 53.9346836937005)	0101000020840800004D4577713F2E_	
7		7 20.0486511452778	54.2857094114533	POINT(20.0486511452778 54.2857094114533)	0101000020840800008DE1C566740C_	
8		8 17.7574020636913	54.7118031530688	POINT(17.7574020636913 54.7118031530688)	010100002084080000217A051AE5C1_	
9		9 20.3702642631934	50.5633591873017	POINT(20.3702642631934 50.5633591873017)	010100002084080000194B85A3C95E_	
10		10 14.701118465429	53.653328947352	POINT(14.701118465429 53.653328947352)	0101000020840800006347DFFFF866_	
11		11 19.2115952007233	50.7024390605123	POINT(19.2115952007233 50.7024390605123)	0101000020840800000B18631A2B36_	
12		12 19.7565805749534	49.9034357458283	POINT(19.7565805749534 49.9034357458283)	010100002084080000B136BA43AFC1_	
13		13 22.2463779096991	51.5219280215223	POINT(22.2463779096991 51.5219280215223)	0101000020840800004A9D689F123F_	
14		14 19.6372935586664	51.1414275231195	POINT(19.6372935586664 51.1414275231195)	010100002084080000766CB0AB25A3_	

Do tabeli została dodana kolumna z geometrią punktów dzięki czemu można zobaczyć podgląd dodanych punktów.



**- proszę stworzyć tabelę województwa i zaimportować tam granice województw**

```
CREATE TABLE public.wojewodztwa
(
    gid integer NOT NULL DEFAULT nextval('wojewodztwa_gid_seq'::regclass),
    jpt_sjr_ko character varying(3) COLLATE pg_catalog."default",
    jpt_kod_je character varying(20) COLLATE pg_catalog."default",
    jpt_nazwa_ character varying(128) COLLATE pg_catalog."default",
    jpt_organ_ character varying(254) COLLATE pg_catalog."default",
    jpt_jor_id integer,
    wersja_od date,
    wersja_do date,
    wazny_od date,
    wazny_do date,
    jpt_kod__1 character varying(3) COLLATE pg_catalog."default",
    jpt_nazwa1 character varying(3) COLLATE pg_catalog."default",
    jpt_organ1 character varying(3) COLLATE pg_catalog."default",
```

```

jpt_wazna_ character varying(3) COLLATE pg_catalog."default",
id_bufora_ numeric,
id_bufora1 numeric,
id_technic integer,
iip_przest character varying(20) COLLATE pg_catalog."default",
iip_identy character varying(128) COLLATE pg_catalog."default",
iip_wersja character varying(32) COLLATE pg_catalog."default",
jpt_kj_iip character varying(20) COLLATE pg_catalog."default",
jpt_kj_i_1 character varying(128) COLLATE pg_catalog."default",
jpt_kj_i_2 character varying(32) COLLATE pg_catalog."default",
jpt_opis character varying(254) COLLATE pg_catalog."default",
jpt_sps_ko character varying(3) COLLATE pg_catalog."default",
id_bufor_1 integer,
jpt_id integer,
jpt_kj_i_3 character varying(3) COLLATE pg_catalog."default",
shape_leng numeric,
shape_area numeric,
geom geometry(MultiPolygon,2180),
CONSTRAINT wojewodztwa_pkey PRIMARY KEY (gid)
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE public.wojewodztwa

```

```

    OWNER to postgres;

```

```

-- Index: wojewodztwa_geom_idx

```

```

-- DROP INDEX public.wojewodztwa_geom_idx;

```

```

CREATE INDEX wojewodztwa_geom_idx

```

```

    ON public.wojewodztwa USING gist

```

```

    (geom)

```

```

    TABLESPACE pg_default;

```

