

Klucz odpowiedzi



support



MINISTERSTWO
ŚRODOWISKA



Sfinansowano ze środków
Narodowego Funduszu
Ochrony Środowiska
i Gospodarki Wodnej

Ćwiczenie 1	2
Ćwiczenie 2	2
Ćwiczenie 3	4
Ćwiczenie 4	4
Ćwiczenie 5	6
Ćwiczenie 6	7
Ćwiczenie 7	7
Ćwiczenie 8	7
Ćwiczenie 9	8
Ćwiczenie 10	8
Ćwiczenie 11	9

Uwaga: dla zadań polegających na zaprojektowaniu zapytania SQL istnieje więcej niż 1 poprawna odpowiedź, niekoniecznie taka sama jak podana w kluczu.

Ćwiczenie 1

1a. TAK - typ MultiPolygon, układ 2180, 1669 obiektów.

1b. spatial_ref_sys, geometry_columns, geography_columns, raster_columns, raster_overviews, gatunki, formuła_e, cdma, siedliska, systematyka

Ćwiczenie 2

2a.

Tabela	Liczba obiektów	Typ geometrii
1.rezerwaty	1669	MultiPolygon
2.gatunki	929	--brak--
3.soo	849	MultiPolygon
4.oso	145	MultiPolygon
5.powiaty	380	MultiPolygon
6.województwa	16	MultiPolygon

2b. Pokazanie nazw i typów danych dla kolumn z tabeli soo:

```
SELECT column_name, data_type FROM information_schema.columns WHERE  
table_name='soo';
```

Pokazanie typu geometrii i układu współrzędnych dla tabeli oso:

```
SELECT type, geometry_type FROM geometry_columns WHERE f_table_name='oso';
```

Ćwiczenie 3

3a.

```
CREATE INDEX ON rezerwaty USING btree(nazwa varchar_pattern_ops);
```

3b.

```
CREATE TABLE lasy_debowe AS SELECT * FROM wydzielenia WHERE species_cd =  
'DB';  
CREATE INDEX ON lasy_debowe USING gist(geom);
```

3c. EXPLAIN (ANALYZE, BUFFERS)

```
3d. SELECT (pg_relation_size('sidx_wydzielenia_geom')::float  
/pg_relation_size('wydzielenia')::float) * 100;
```

Ćwiczenie 4

4a. Oczekiwany wynik: zwrócenie wszystkich kolumn z tabeli wydzielenia, gdzie spec_age przekracza 100 i species_cd to DB lub BK.

```
SELECT * FROM wydzielenia WHERE spec_age > 100 AND species_cd IN ('DB','BK');
```

Oczekiwany wynik: zwrócenie takich wierszy z tabeli wydzielenia, gdzie species_cd to BRZ, SO lub jest wartością NULL.

```
SELECT * FROM wydzielenia WHERE species_cd IN ('BRZ','SO') OR species_cd IS NULL;
```

Oczekiwany wynik: zwrócenie takich wierszy z tabeli pomniki_przyrody, gdzie atrybut "gatunek" zawiera słowo "dąb" bez uwzględnienia wielkości liter, a data_utwor zawiera się w przedziale od 13.01.2002 do 13.12.2007.

```
SELECT * FROM pomniki_przyrody WHERE gatunek ILIKE 'dąb%' AND data_utwor  
BETWEEN '2002-01-13'::date AND '2007-12-13'::date;
```

Oczekiwany wynik: zwrócenie takich wierszy z tabeli pomniki_przyrody, które są położone w powiecie głubczyckim.

```
SELECT a.* FROM pomniki_przyrody a, powiaty b WHERE ST_Within(a.geom, b.geom)
AND b.jpt_nazwa_ = 'powiat głubczycki';
```

Oczekiwany wynik: zwrócenie takich wierszy z tabeli drogi, które są położone w powiecie nyskim i nie przekraczają jego granicy.

```
SELECT a.* FROM drogi a, powiaty b WHERE ST_Within(a.geom, b.geom) AND
b.jpt_nazwa_ = 'powiat nyski';
```

Oczekiwany wynik: znajdź 5 wydzieleń o najwyższym wieku (spec_age) które nie są położone częściowo lub całkowicie na terenie rezerwatu przyrody.

```
SELECT DISTINCT a.adr_for, a.spec_age FROM wydzielenia a, rezerwaty b WHERE
ST_Intersects(a.geom, b.geom) = FALSE ORDER BY spec_age DESC LIMIT 5;
```

Oczekiwany wynik: przypisz do każdej miejscowości w powiecie nyskim pomnik przyrody, który będzie najbliższym centrum miejscowości.

```
SELECT DISTINCT ON(a.naz_glowna) a.naz_glowna, b.nazwa, b.gatunek, b.obiekt,
b.kodinspire
FROM miejscowosci a, pomniki_przyrody b WHERE a.powiat = 'nyski' ORDER BY
a.naz_glowna, a.geom <-> b.geom
```

Oczekiwany wynik: znajdź wszystkie dęby - pomniki przyrody w promieniu 10 km od najstarszego drzewostanu dębowego w tabeli wydzielenia.

```
SELECT a.* FROM pomniki_przyrody a, wydzielenia b
WHERE st_dwithin(a.geom, b.geom, 10000)
AND b.species_cd = 'DB'
AND a.gatunek ILIKE 'dąb%'
AND b.spec_age = (SELECT max(spec_age) FROM wydzielenia WHERE species_cd =
'DB');
```

4b.

A-2, B-3, C-4, D-1

Ćwiczenie 5

5a. A - 2, B - 1, C - 3

5b.

Oczekiwany wynik: oblicz powierzchnię dla każdego rezerwatu.

```
SELECT nazwa, kodinspire, ST_Area(geom) FROM rezerwaty;
```

Oczekiwany wynik: narysuj strefę buforową 500 m od każdego pomnika przyrody.

```
SELECT nazwa, gatunek, kodinspire, ST_Buffer(geom, 500) FROM pomniki_przyrody;
```

Oczekiwany wynik: oblicz powierzchnię (w hektarach) i obwód części wspólnej dla par obszarów OSO i SOO dla wszystkich przypadków, gdzie obszary te się pokrywają.

```
SELECT ST_Area(ST_Intersection(a.geom, b.geom)) / 1000, a.nazwa AS nazwa_oso,  
b.nazwa AS nazwa_soo  
FROM oso a, soo b WHERE ST_Overlaps(a.geom, b.geom);
```

5c.

```
SELECT * FROM wydzielena WHERE ST_Within(st_centroid(geom), geom) = FALSE;
```

```
SELECT a.adr_for, ST_Difference(a.geom, b.geom) AS geom FROM wydzielena a,  
rezerwaty b  
WHERE ST_Overlaps(a.geom, b.geom);
```

5d.

```
SELECT a.adr_for, sum(ST_Length(ST_Intersection(a.geom, b.geom)))  
FROM wydzielena a, drogi b  
WHERE ST_Intersects(a.geom, b.geom)  
GROUP BY a.adr_for;
```

Ćwiczenie 6

- 6a. NIE, ponieważ jest to tylko ROLE bez prawa LOGIN.
- 6b. NIE
- 6c. NIE
- 6d. GRANT ALL ON zbiorniki_wodne TO sosna;
- 6e. GRANT INSERT, UPDATE ON zbiorniki_wodne TO sosna;

Ćwiczenie 7

7a.
BEGIN;
CREATE TABLE drzewa(id serial, gatunek varchar);
INSERT INTO drzewa(gatunek) VALUES('sosna');
INSERT INTO drzewa(gatunek) VALUES('brzoza');
INSERT INTO drzewa(gatunek) VALUES('dąb');
INSERT INTO drzewa(gatunek) VALUES('jodła');
COMMIT;

7b.
BEGIN;
CREATE TABLE drzewa(id serial, gatunek varchar);
SAVEPOINT tabela;
INSERT INTO drzewa(gatunek) VALUES('sosna');
INSERT INTO drzewa(gatunek) VALUES('brzoza');
INSERT INTO drzewa(gatunek) VALUES('dąb');
INSERT INTO drzewa(gatunek) VALUES('jodła');
COMMIT;

7c. ALTER TABLE siedliska ADD COLUMN IF NOT EXISTS zbiorowisko VARCHAR;

7d. ALTER TABLE gatunki RENAME COLUMN latin_name TO nazwa_lacinska;

Ćwiczenie 8

```
8a. SELECT jpt_nazwa_ AS powiat, powierzchnia_rez/1000000 AS powierzchnia_rez ,  
(powierzchnia_rez / powierzchnia_pow)*100 AS procent_rez  
FROM (  
  SELECT a.jpt_nazwa_ , sum(  
    CASE WHEN ST_Overlaps(a.geom,b.geom) THEN  
  ST_Area(ST_Intersection(a.geom,b.geom))  
    WHEN ST_Contains(a.geom, b.geom) THEN ST_Area(b.geom)  
    ELSE 0 END  
  )  
  AS powierzchnia_rez, ST_Area(a.geom) AS powierzchnia_pow FROM powiaty a,  
rezerwy b  
WHERE ST_Intersects(a.geom, b.geom)  
GROUP BY a.id) raport  
ORDER BY powiat;
```

Ćwiczenie 9

```
9a. SELECT * FROM rezerwy WHERE nazwa ~ '(na|przy|w) .';  
9b. SELECT kod, regexp_replace(kod, '[:alpha:]', ',' , 'g') FROM soo;
```

Ćwiczenie 10

```
10a. WITH zbiornik AS  
(SELECT geom FROM oso WHERE nazwa = 'Zbiornik Nyski')  
SELECT a.* FROM drogi a, zbiornik b WHERE ST_Intersects(a.geom, b.geom);
```

```
10b. SELECT site_type CASE  
WHEN site_type like 'B%' THEN 'bór'  
WHEN site_type like 'L%' OR site_type like 'O%' THEN 'las'  
WHEN site_type IS NULL THEN 'teren nieleśny'  
ELSE 'pozostałe' END AS typ FROM wydzielenia;
```

```
10c. SELECT nazwa, kodinspire, CASE  
WHEN ST_Area(geom) > 1000000 THEN ST_Simplify(geom, 50)
```



```
ELSE ST_PointOnSurface(geom) END AS geom FROM rezerwaty;
```

```
10d. WITH bufor_drog AS  
(SELECT CASE WHEN highway='track' THEN ST_Buffer(geom, 5) ELSE ST_Buffer(geom,  
10) END AS geometry FROM drogi)  
SELECT a.adr_for, coalesce(ST_Area(ST_Union(ST_Difference(a.geom,  
b.geometry))),max(ST_Area(a.geom)))  
FROM wydzielania a LEFT OUTER JOIN bufor_drog b  
ON ST_Intersects(a.geom, b.geometry)  
GROUP BY a.adr_for;
```

Ćwiczenie 11

```
11a. CREATE VIEW lasy_bukowe AS SELECT * FROM wydzielania WHERE species_cd =  
'BK';
```

```
11b. CREATE MATERIALIZED VIEW natura_gen50 AS  
SELECT nazwa, kod, kodinspire, ST_Simplify(geom, 50) FROM soo  
UNION ALL  
SELECT nazwa, kod, kodinspire, ST_Simplify(geom, 50) FROM oso;
```

```
11c.
```

```
REFRESH MATERIALIZED VIEW natura_gen50;
```