

Raport

Przetwarzanie i analiza danych przestrzennych

Oracle spatial

Jacek Budny, Mateusz Kleszcz

Celem ćwiczenia jest zapoznanie się ze sposobem przechowywania, przetwarzania i analizy danych przestrzennych w bazach danych
(na przykładzie systemu Oracle spatial)

Swoje odpowiedzi wpisuj w miejsca oznaczone jako:

Wyniki, zrzut ekranu, komentarz

-- ...

Do wykonania ćwiczenia (zadania 1 – 7) i wizualizacji danych wykorzystaj Oracle SQL Developer.
Alternatywnie możesz wykonać analizy w środowisku Python/Jupyter Notebook

Do wykonania zadania 8 wykorzystaj środowisko Python/Jupyter Notebook

Raport należy przesłać w formacie pdf.

Należy też dołączyć raport zawierający kod w formacie źródłowym.

Np.

- plik tekstowy .sql z kodem poleceń
- plik .md zawierający kod wersji tekstowej
- notebook programu jupyter – plik .ipynb

Zamieść kod rozwiązania oraz zrzuty ekranu pokazujące wyniki, (dołącz kod rozwiązania w formie tekstowej/źródłowej)

Zwróć uwagę na formatowanie kodu

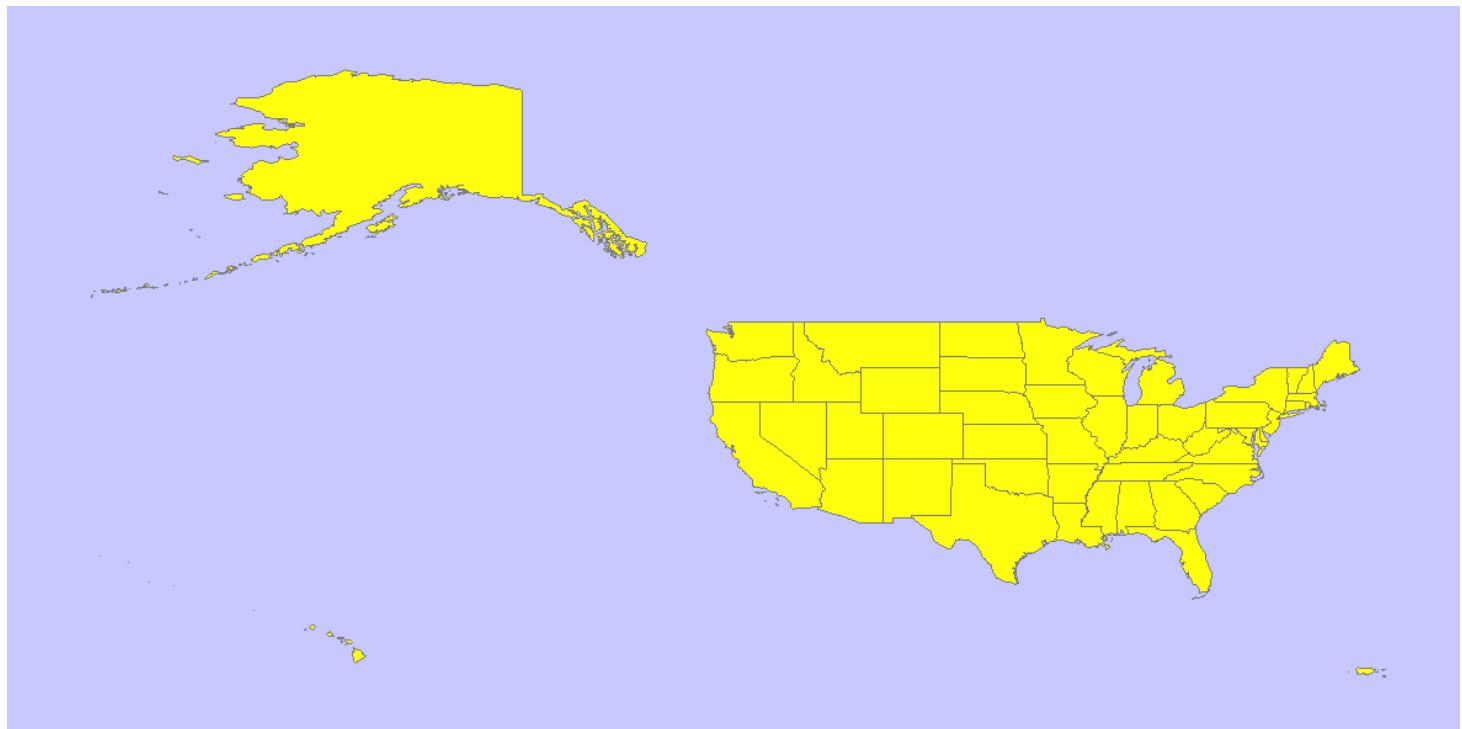
Zadanie 1

Zwizualizuj przykładowe dane

US_STATES

Wyniki, zrzut ekranu, komentarz

```
SELECT * FROM US_STATES
```

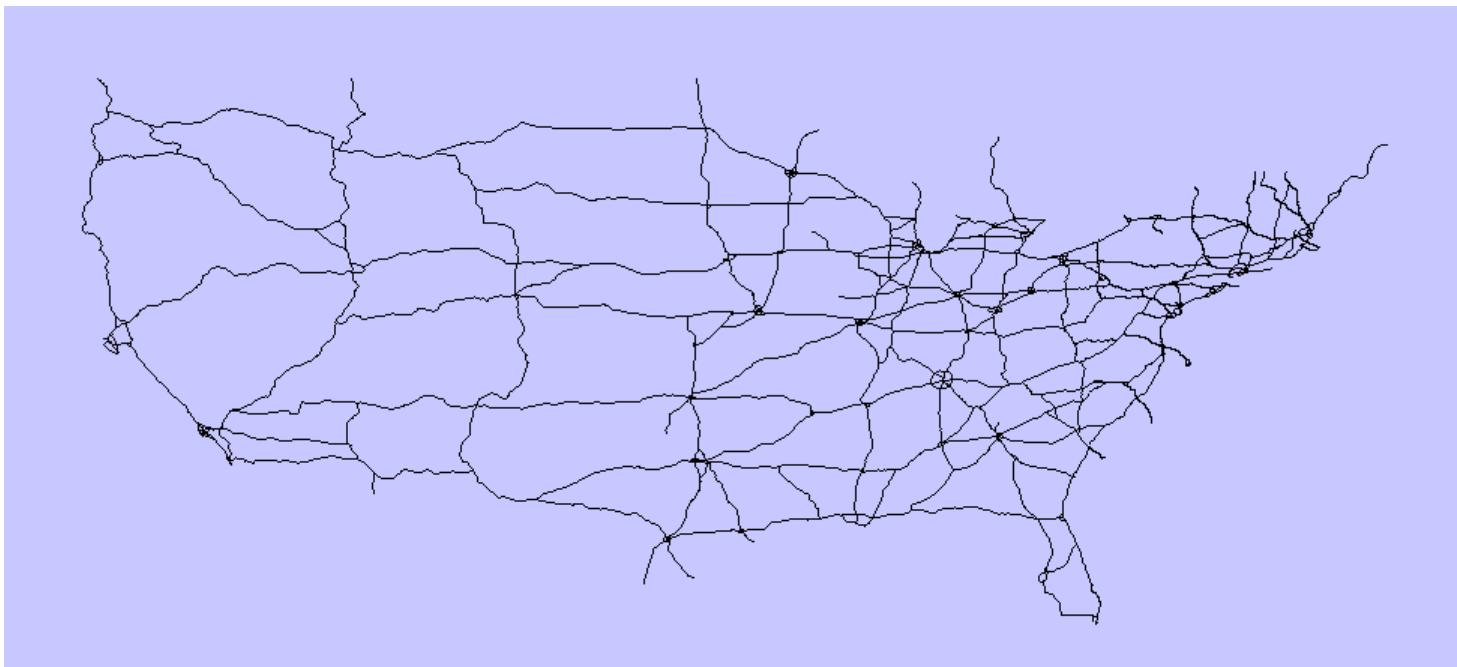
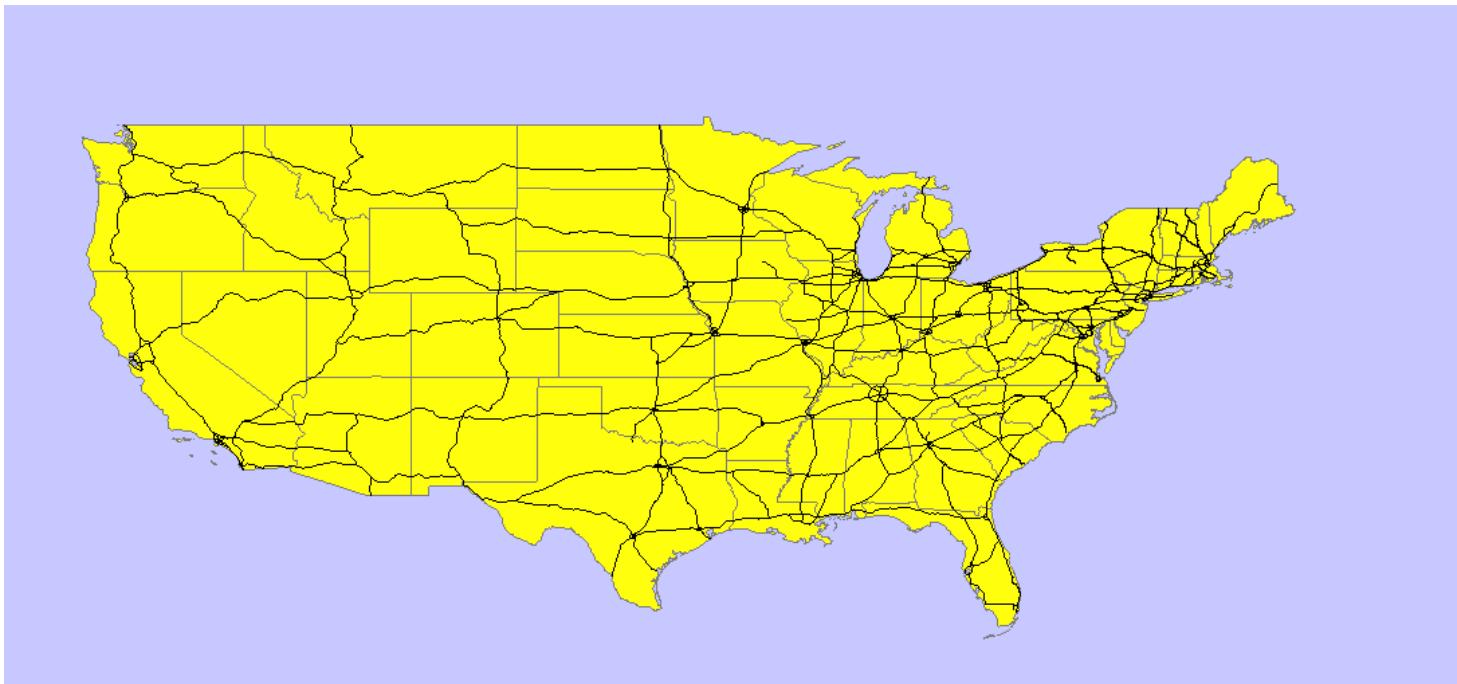


Wizualizacja przedstawia stany USA. Tabela ze stanami zawiera kolumny z opisem geometrycznym.

US_INTERSTATES

Wyniki, zrzut ekranu, komentarz

```
SELECT * FROM US_INTERSTATES
```



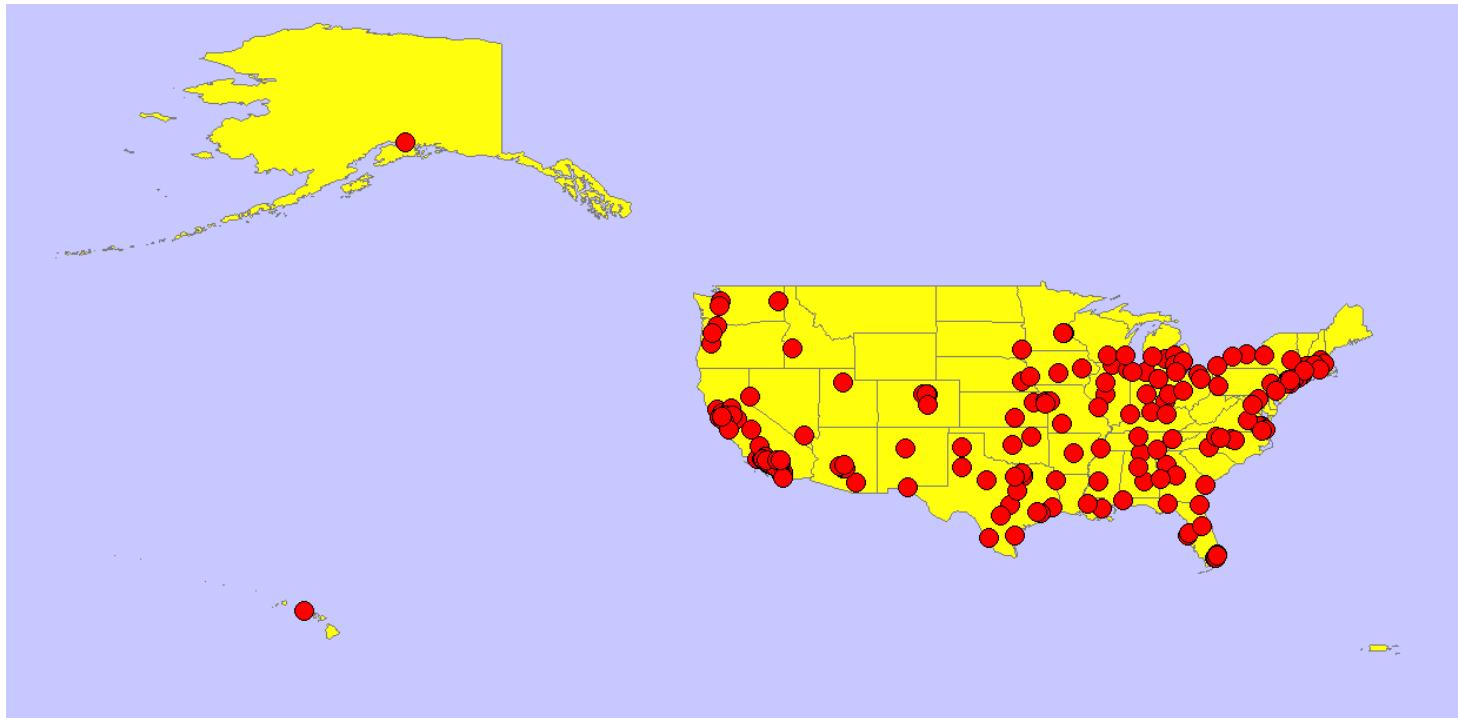
	ID	INTERSTATE	GEOM
1	32	I229	[MDSYS.SDO_GEOMETRY]
2	33	I235	[MDSYS.SDO_GEOMETRY]

W tabeli przedstawione są drogi stanów zjednoczonych (nie ma żadnych dróg z Alaski i wysp, tylko z kontynentu). Tabela zawiera ID, nazwę oraz opis geometryczny.

US_CITIES

Wyniki, zrzut ekranu, komentarz

```
SELECT * FROM US_CITIES
```



SQL | All Rows Fetched: 195 in 0.128 seconds

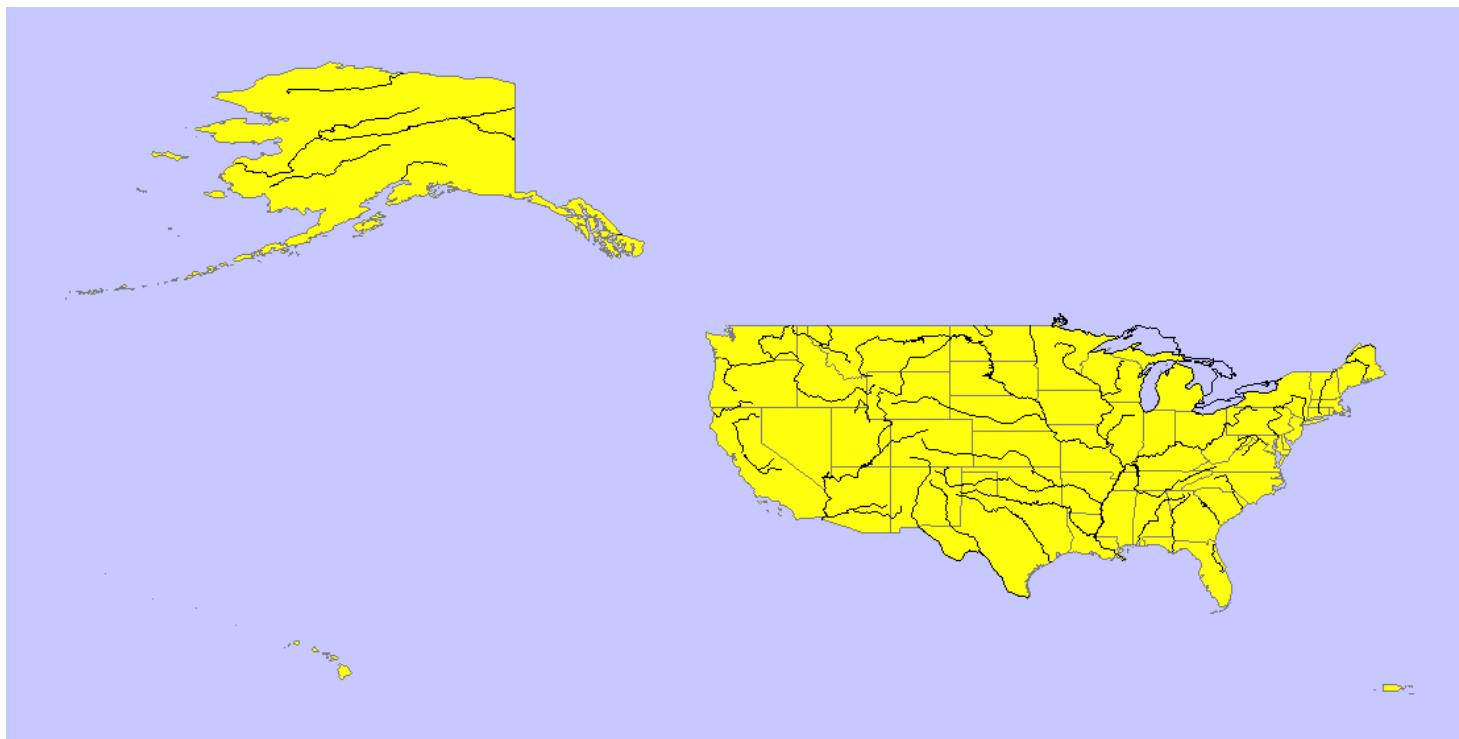
	ID	CITY	STATE_ABRV	POP90	RANK90	LOCATION
1	139	Scottsdale	AZ	130069	139	[MDSYS.SDO_Geometry]
2	140	Plano	TX	128713	140	[MDSYS.SDO_Geometry]

Tabela ta zawiera 195 miast, każde jest przypisane do stanu oraz posiada swoją punktową lokalizację.

US_RIVERS

Wyniki, zrzut ekranu, komentarz

```
SELECT * FROM US_RIVERS
```



SQL | All Rows Fetched: 56 in 0.138 seconds

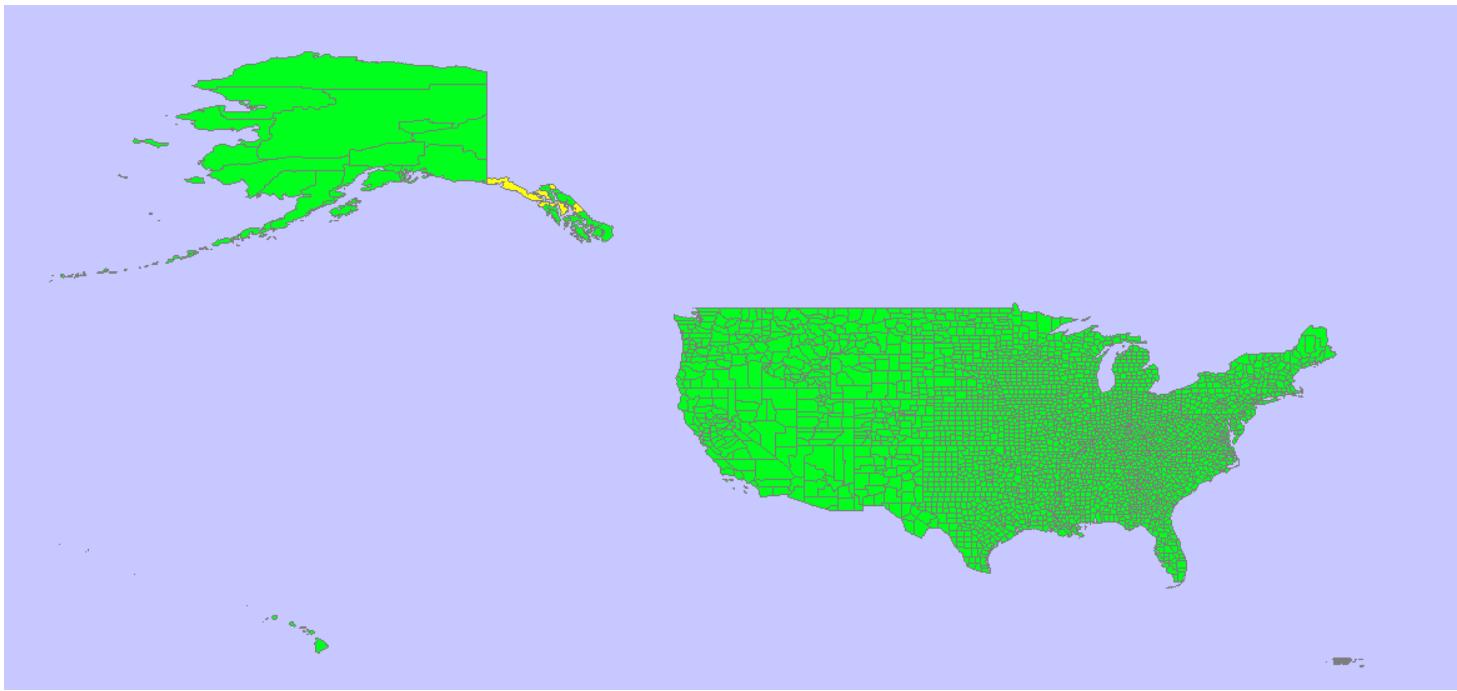
ID	NAME	SYSTEM	GEOM
51	51 Koyukuk	Yukon	[MDSYS.SDO_GEOMETRY]
52	52 Kuskokwim	(null)	[MDSYS.SDO_GEOMETRY]

Tabela przedstawia 56 rzek w USA, są one wizualizowane w ten sam sposób co autostrady

US_COUNTIES

Wyniki, zrzut ekranu, komentarz

```
SELECT * FROM US_COUNTIES
```



SQL | All Rows Fetched: 3230 in 1.96 seconds

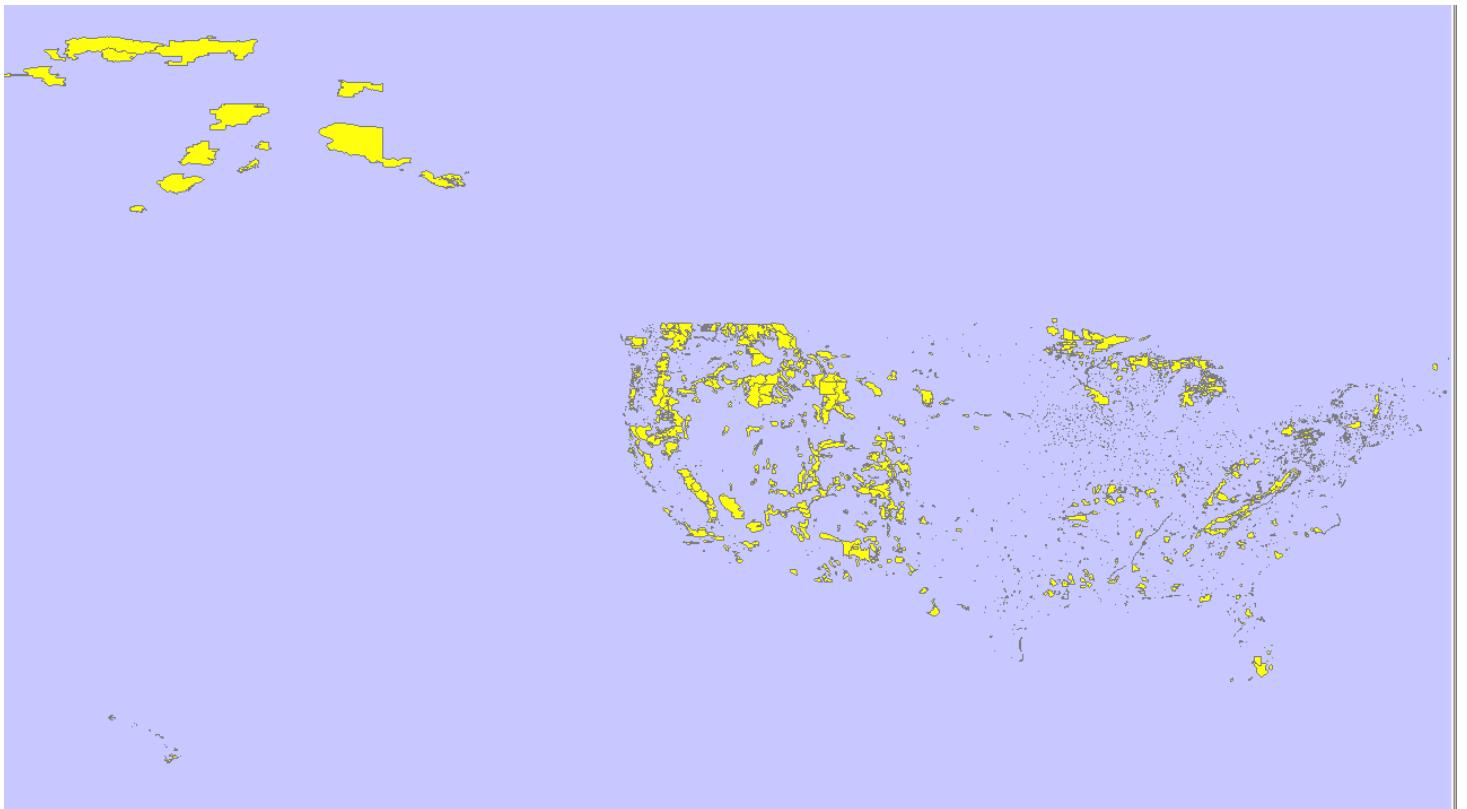
ID	COUNTY	FIPSSTCO	STATE	STATE_ABRV	FIPSST	LANDSQMI	TOTPOP	POPPSQMI	GEOM
3225	2942	Richmon...	51760	Virginia	VA	51	60,1163	203056	3377,7167 [MDSYS.SDO_GEOMETRY]
3226	2943	Roanoke...	51770	Virqinia	VA	51	42,8941	96397	2247,3203 [MDSYS.SDO GEOMETRY]

Obszar USA podzielony na hrabstwa, przed wykonaniem query otrzymujemy warning o możliwym dużym zużyciu zasobów (3230 wierszy, każdy z nich ma swoją geometrię)

US_PARKS

Wyniki, zrzut ekranu, komentarz

```
SELECT * FROM US_PARKS
```



Script Output x Query Result x

SQL | All Rows Fetched: 6331 in 5.464 seconds

	ID	NAME	FCC	GEOM
6326	5502	Interstate	D85	[MDSYS.SDO_GEOMETRY]
6327	5503	Interstate	D85	[MDSYS.SDO_GEOMETRY]
6328	5504	Sibley	D85	[MDSYS.SDO_GEOMETRY]
6329	5505	Monson Lake	D85	[MDSYS.SDO_GEOMETRY]
6330	5506	Lake Maria	D85	[MDSYS.SDO_GEOMETRY]
6331	5507	William O'Brien	D85	[MDSYS.SDO_GEOMETRY]

Tabela zawiera wszystkie parki w USA, jest to największa tabela (6331 wiersze), każdy park jest reprezentowany przez geometrię małych rozmiarów.

Zadanie 2

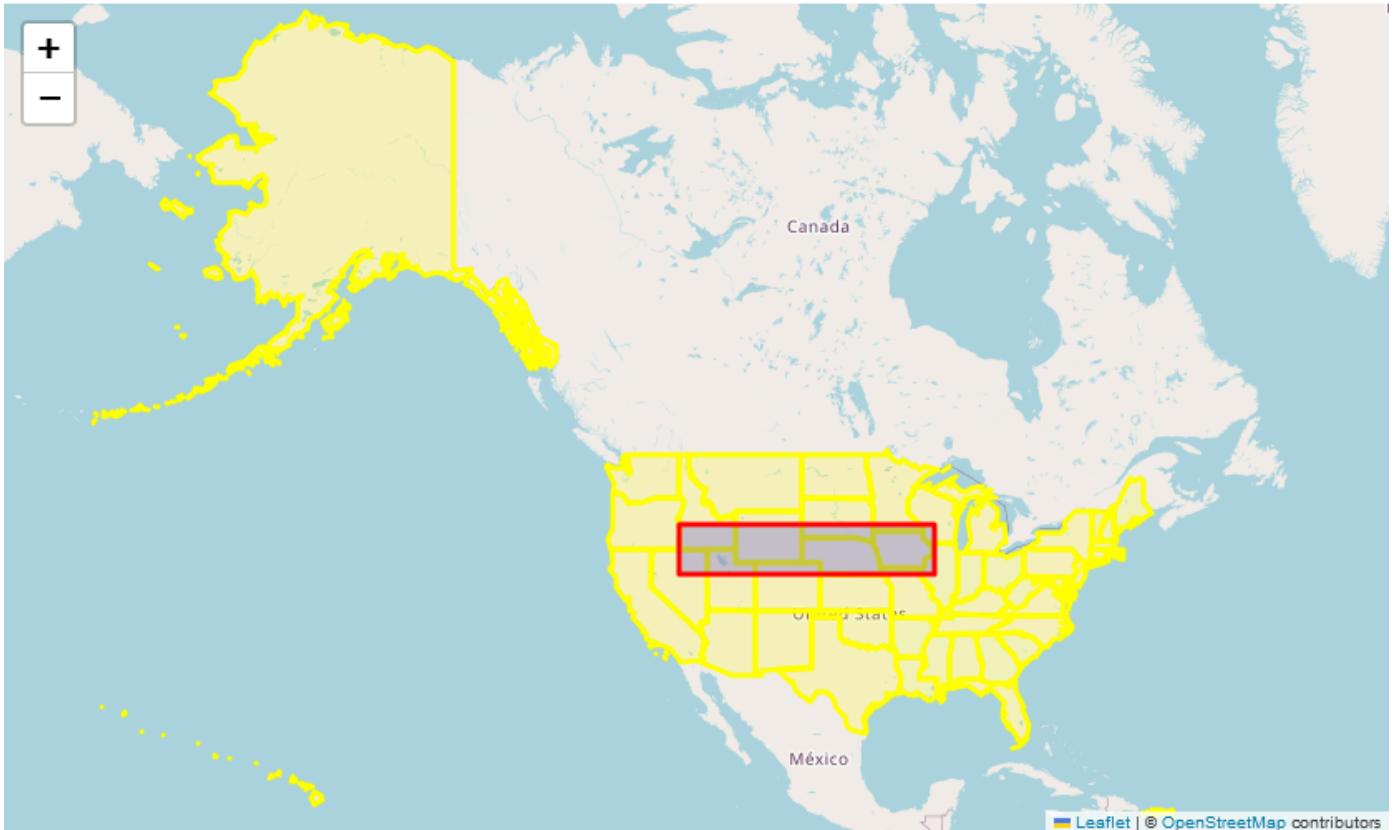
Znajdź wszystkie stany (us_states) których obszary mają część wspólną ze wskazaną geometrią (prostokątem)

Pokaż wynik na mapie.

prostokąt

```
SELECT sdo_geometry (2003, 8307, null,
sdo_elem_info_array (1,1003,3),
sdo_ordinate_array ( -117.0, 40.0, -90., 44.0)) g
FROM dual
```

Wyniki, zrzut ekranu, komentarz



Widok prostokąta na tle stanów.

```
# stany
SELECT sdo_util.to_wktgeometry(geom)
FROM us_states

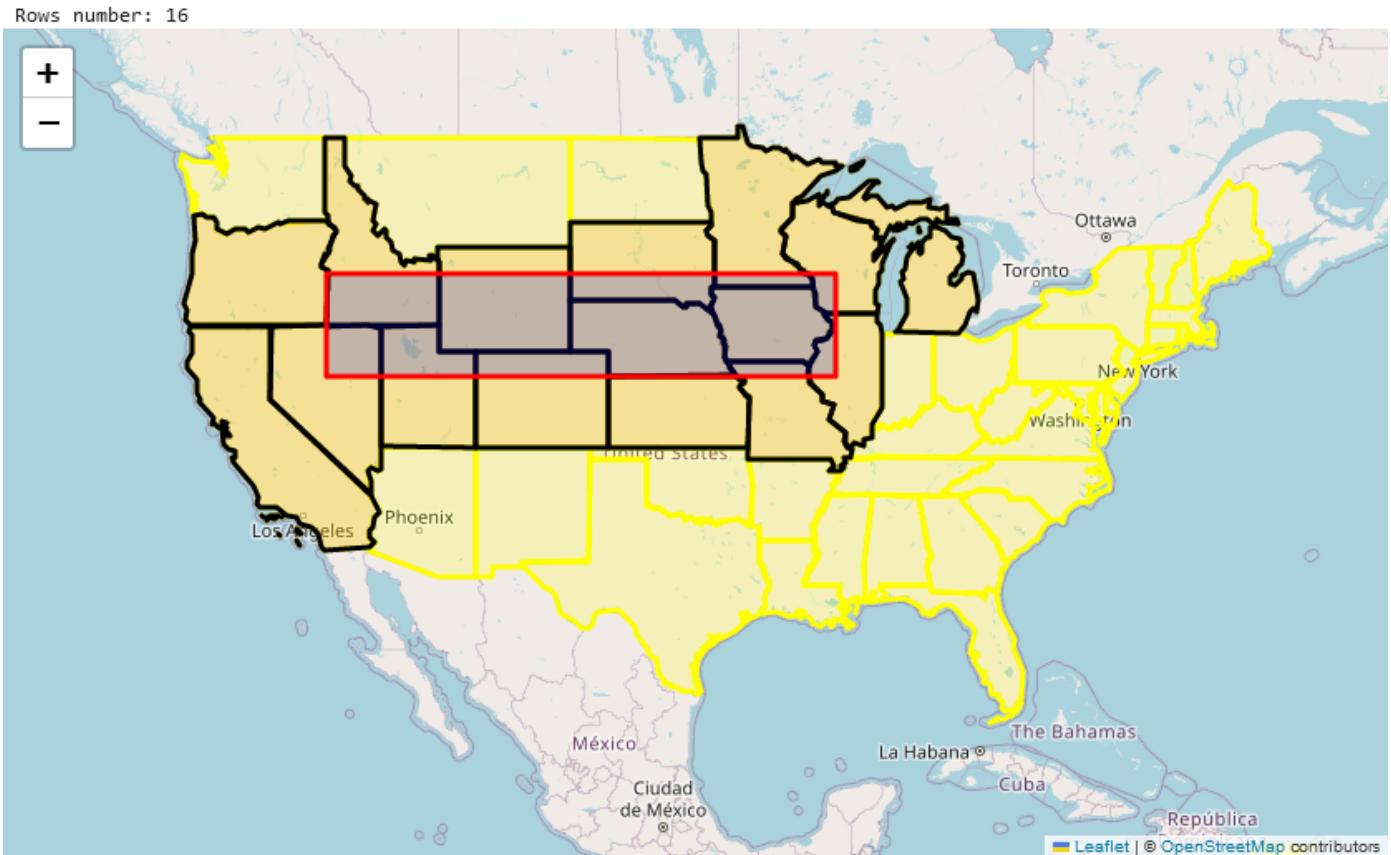
# prostokąt
SELECT sdo_util.to_wktgeometry(
  sdo_geometry(
    2003, 8307, null,
    sdo_elem_info_array (1,1003,3),
    sdo_ordinate_array ( -117.0, 40.0, -90., 44.0)
  )
) g
FROM dual
```

Użyj funkcji SDO_FILTER

```
SELECT state, geom FROM us_states
WHERE sdo_filter (geom,
sdo_geometry (2003, 8307, null,
sdo_elem_info_array (1,1003,3),
sdo_ordinate_array ( -117.0, 40.0, -90., 44.0))
) = 'TRUE';
```

Zwróć uwagę na liczbę zwróconych wierszy (16)

Wyniki, zrzut ekranu, komentarz



Funkcja SDO_FILTER zwraca również stany, które nie mają części wspólnej z prostokątem. Operator ten wykonuje tylko podstawową operację filtrowania, odpowiada na pytanie, czy geometria1 może mieć część wspólną z geometrią2. Stany postrzegane są jako prostokąty przez tę metodę.

```
# stany z częścią wspólną
SELECT sdo_util.to_wktgeometry(geom) FROM us_states
WHERE sdo_filter (
    geom,
    sdo_geometry (
        2003, 8307, null,
        sdo_elem_info_array (1,1003,3),
        sdo_ordinate_array ( -117.0, 40.0, -90., 44.0)
    )
) = 'TRUE'
```

Użyj funkcji SDO_ANYINTERACT

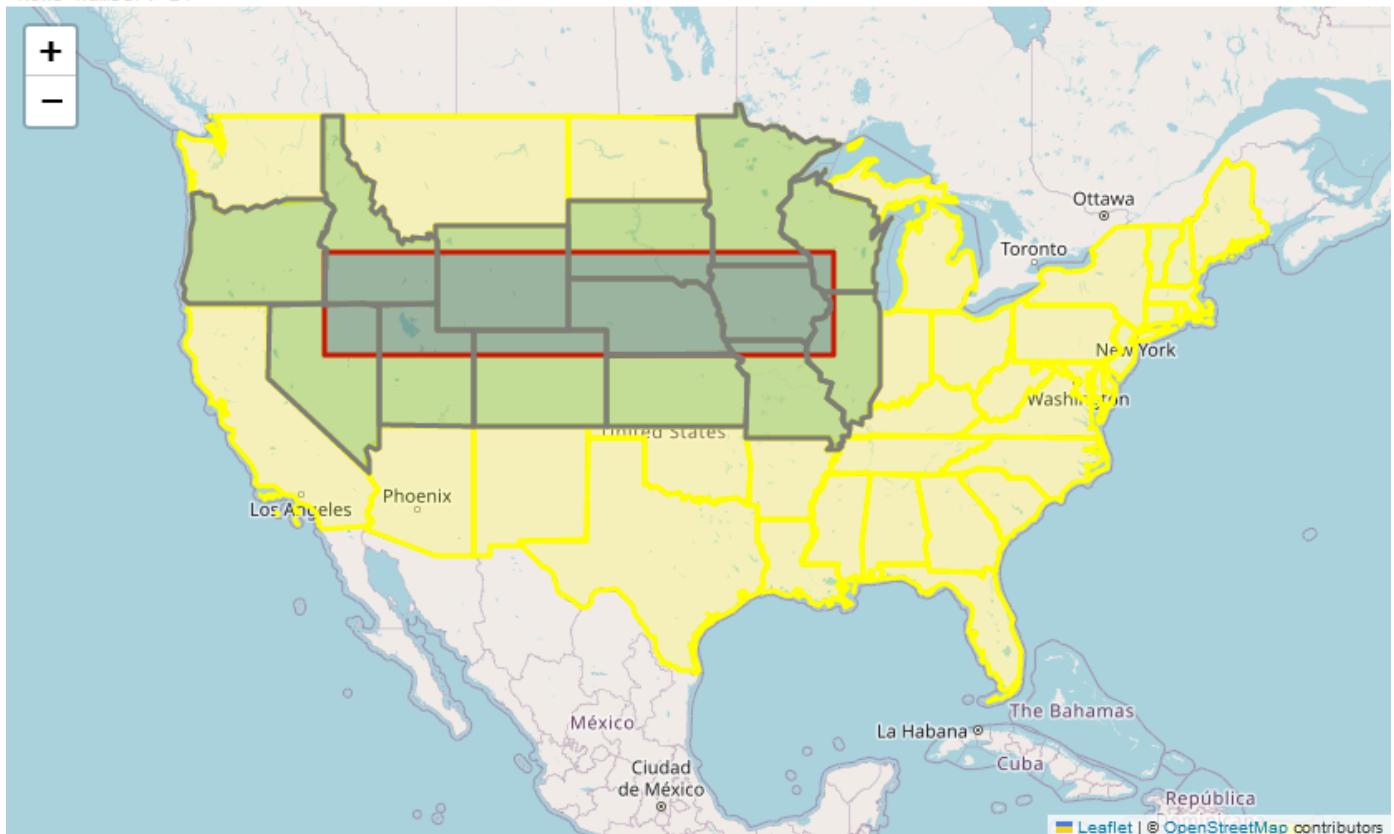
```
SELECT state, geom FROM us_states
WHERE sdo_anyinteract (geom,
    sdo_geometry (2003, 8307, null,
        sdo_elem_info_array (1,1003,3),
        sdo_ordinate_array ( -117.0, 40.0, -90., 44.0))
) = 'TRUE';
```

Porównaj wyniki sdo_filter i sdo_anyinteract

Pokaż wynik na mapie

Wyniki, zrzut ekranu, komentarz

Rows number: 14



SDO_ANYINTERACT sprawdza, czy jakiekolwiek geometrie w tabeli mają relację topologiczną ANYINTERACT z określoną geometrią. Jest to równoważne określeniu operatora SDO_RELATE z "mask=ANYINTERACT". SDO_RELATE z tą maską odpowiada na pytanie, czy geometria1 posiada część wspólną z geometrią2. Pod uwagę brane są stany które rzeczywiście mają część wspólną z prostokątem, dlatego liczba zwracanych stanów jest mniejsza.

```
SELECT sdo_util.to_wktgeometry(geom) FROM us_states
WHERE sdo_anyinteract (
    geom,
    sdo_geometry (2003, 8307, null,
        sdo_elem_info_array (1,1003,3),
        sdo_ordinate_array ( -117.0, 40.0, -90., 44.0)
    )
) = 'TRUE'
```

Zadanie 3

Znajdź wszystkie parki (us_parks) których obszary znajdują się wewnątrz stanu Wyoming

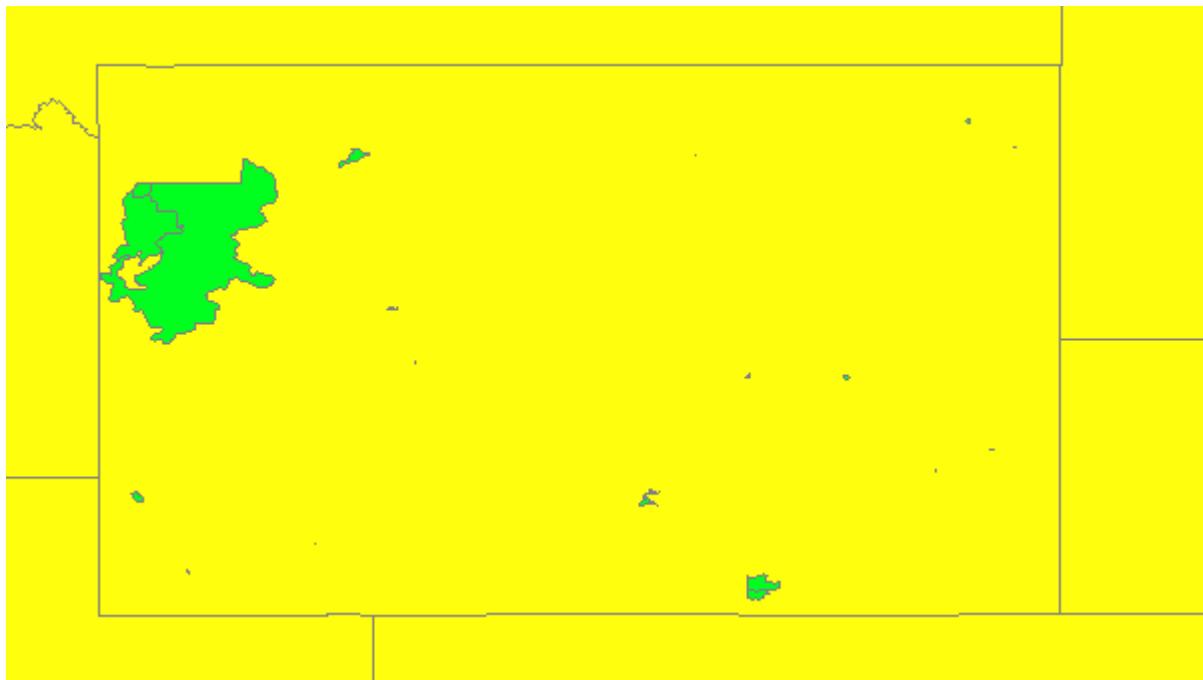
Użyj funkcji SDO_INSIDE

```
SELECT p.name, p.geom
FROM us_parks p, us_states s
WHERE s.state = 'Wyoming'
AND SDO_INSIDE (p.geom, s.geom) = 'TRUE';
```

W przypadku wykorzystywania narzędzia SQL Developer, w celu wizualizacji na mapie użyj podzapytania

```
SELECT pp.name, pp.geom FROM us_parks pp
WHERE id IN
(
  SELECT p.id
  FROM us_parks p, us_states s
  WHERE s.state = 'Wyoming'
  and SDO_INSIDE (p.geom, s.geom) = 'TRUE'
)
```

Wyniki, zrzut ekranu, komentarz



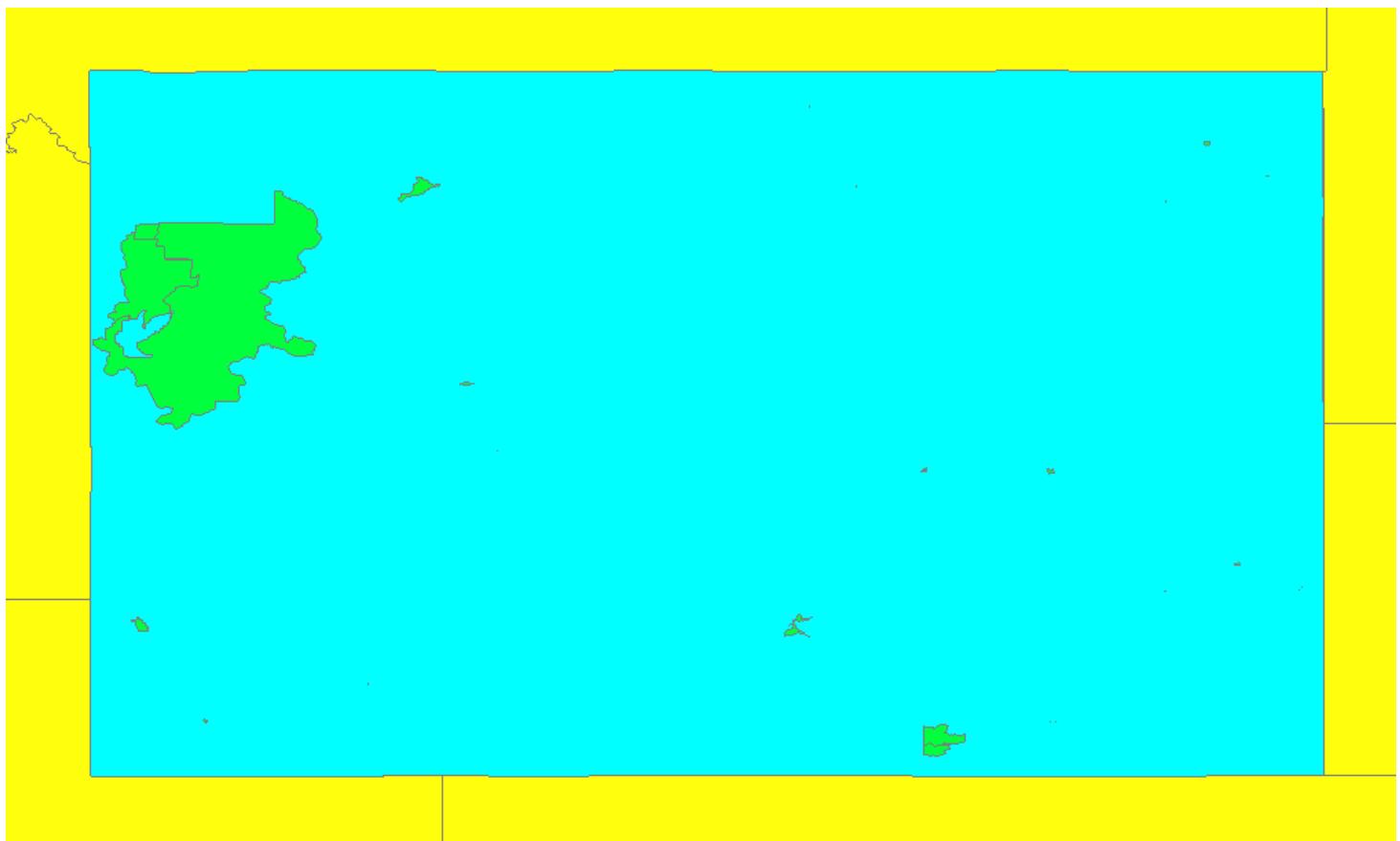
 SQL | All Rows Fetched: 32 in 0.289 seconds

	NAME	GEOM
1	Flume Creek Park	[MDSYS.SDO_GEOMETRY]
2	Cinnabar Park	[MDSYS.SDO_GEOMETRY]
3	Sinclair Recreation Park	[MDSYS.SDO_GEOMETRY]
4	Kendrick Park	[MDSYS.SDO_GEOMETRY]
5	Beartrap Meadow County Park	[MDSYS.SDO_GEOMETRY]

| Na mapie widzimy 32 parki, które w całości znajdują się bezpośrednio w stanie Wyoming USA.

```
SELECT state, geom FROM us_states  
WHERE state = 'Wyoming'
```

| Wyniki, zrzut ekranu, komentarz



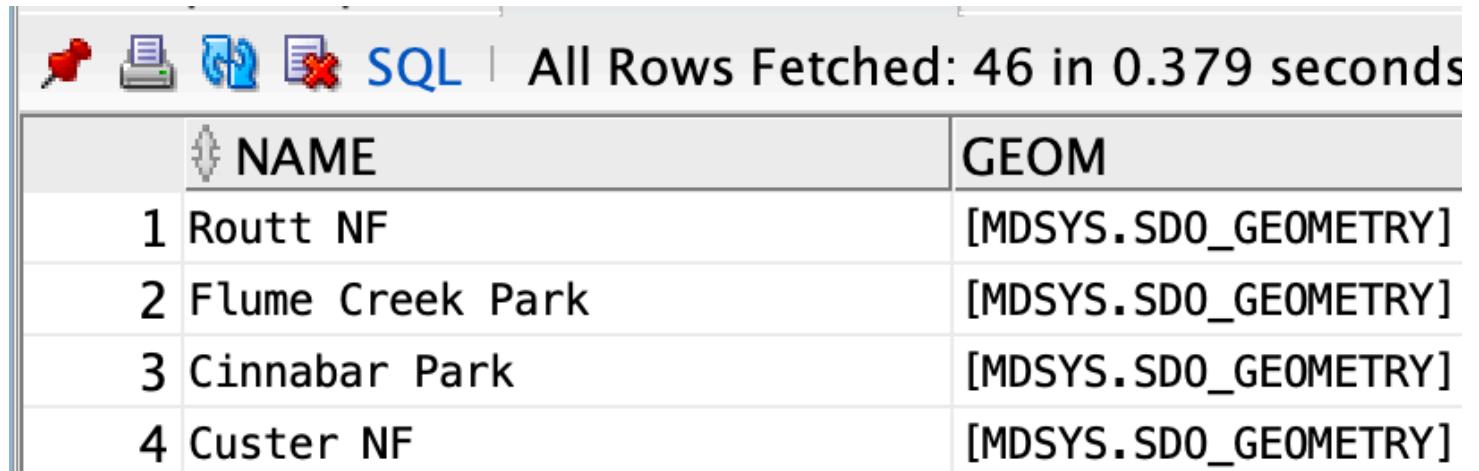
| Dzięki temu poleceniu możemy oznaczyć stan Wyoming na mapie USA.

Porównaj wynik z:

```
SELECT p.name, p.geom  
FROM us_parks p, us_states s  
WHERE s.state = 'Wyoming'  
AND SDO_ANYINTERACT (p.geom, s.geom) = 'TRUE';
```

W celu wizualizacji użyj podzapytania

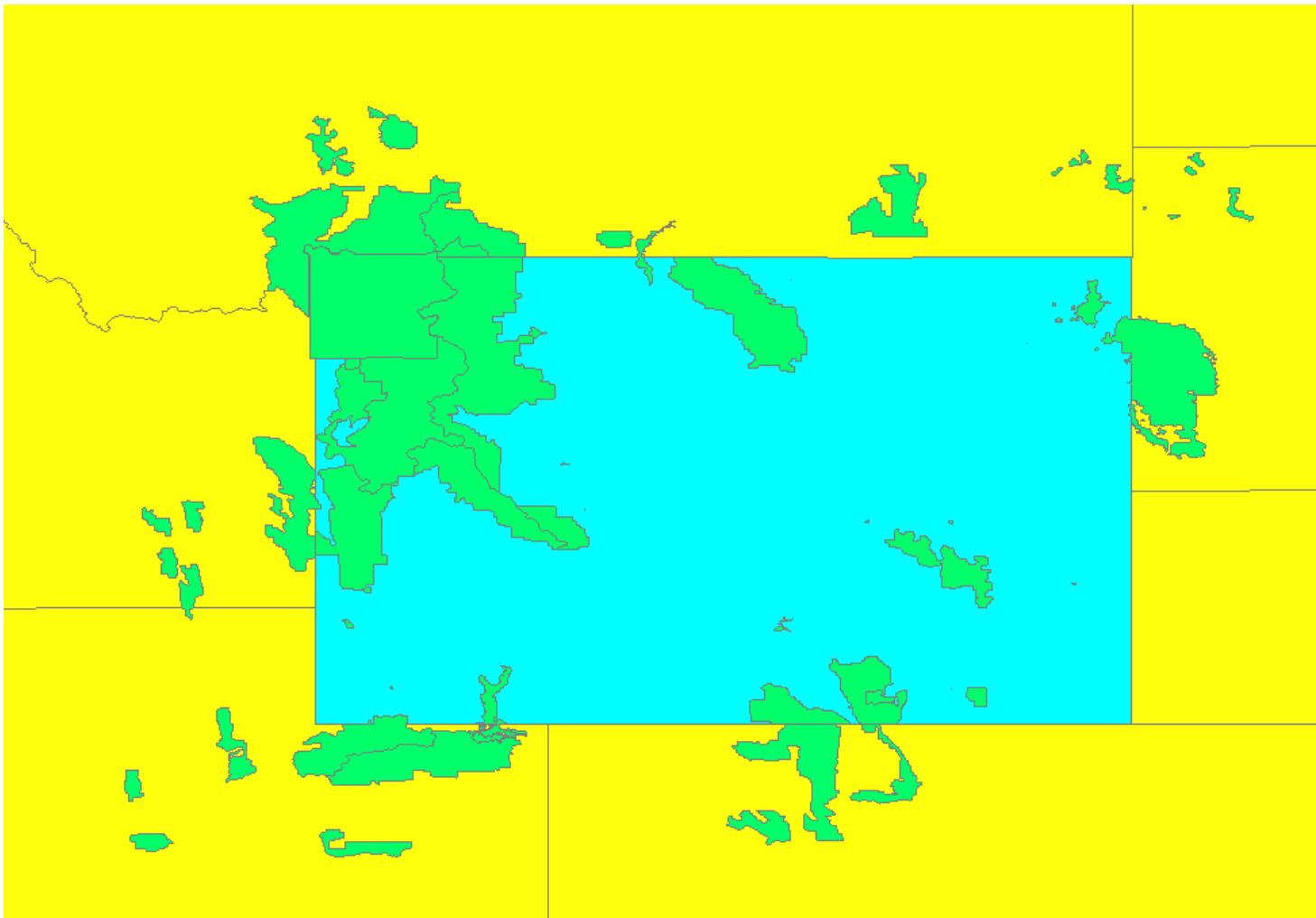
Wyniki, zrzut ekranu, komentarz



The screenshot shows a SQL query results window. At the top, there are icons for refresh, print, and export, followed by the word "SQL". To the right of the SQL icon, it says "All Rows Fetched: 46 in 0.379 seconds". The main area is a table with two columns: "NAME" and "GEOM". The "NAME" column contains four rows labeled 1, 2, 3, and 4, corresponding to park names. The "GEOM" column contains four entries, all of which are "[MDSYS.SDO_GEOMETRY]".

	NAME	GEOM
1	Routt NF	[MDSYS.SDO_GEOMETRY]
2	Flume Creek Park	[MDSYS.SDO_GEOMETRY]
3	Cinnabar Park	[MDSYS.SDO_GEOMETRY]
4	Custer NF	[MDSYS.SDO_GEOMETRY]

```
SELECT pp.name, pp.geom FROM us_parks pp  
WHERE id IN  
(  
    SELECT p.id  
    FROM us_parks p, us_states s  
    WHERE s.state = 'Wyoming'  
    and SDO_ANYINTERACT (p.geom, s.geom) = 'TRUE'  
)
```



Tym razem otrzymaliśmy 46 wierszy. Drugie użyte przez nas zapytanie zwraca parki, których jakakolwiek część leży w granicach stanu Wyoming, podczas gdy w pierwszym park musi zawierać się w całości w granicach.

Zadanie 4

Znajdź wszystkie jednostki administracyjne (us_counties) wewnątrz stanu New Hampshire

```
SELECT c.county, c.state_abrv, c.geom  
FROM us_counties c, us_states s  
WHERE s.state = 'New Hampshire'  
AND SDO_RELATE ( c.geom,s.geom, 'mask=INSIDE+COVEREDBY') = 'TRUE';
```

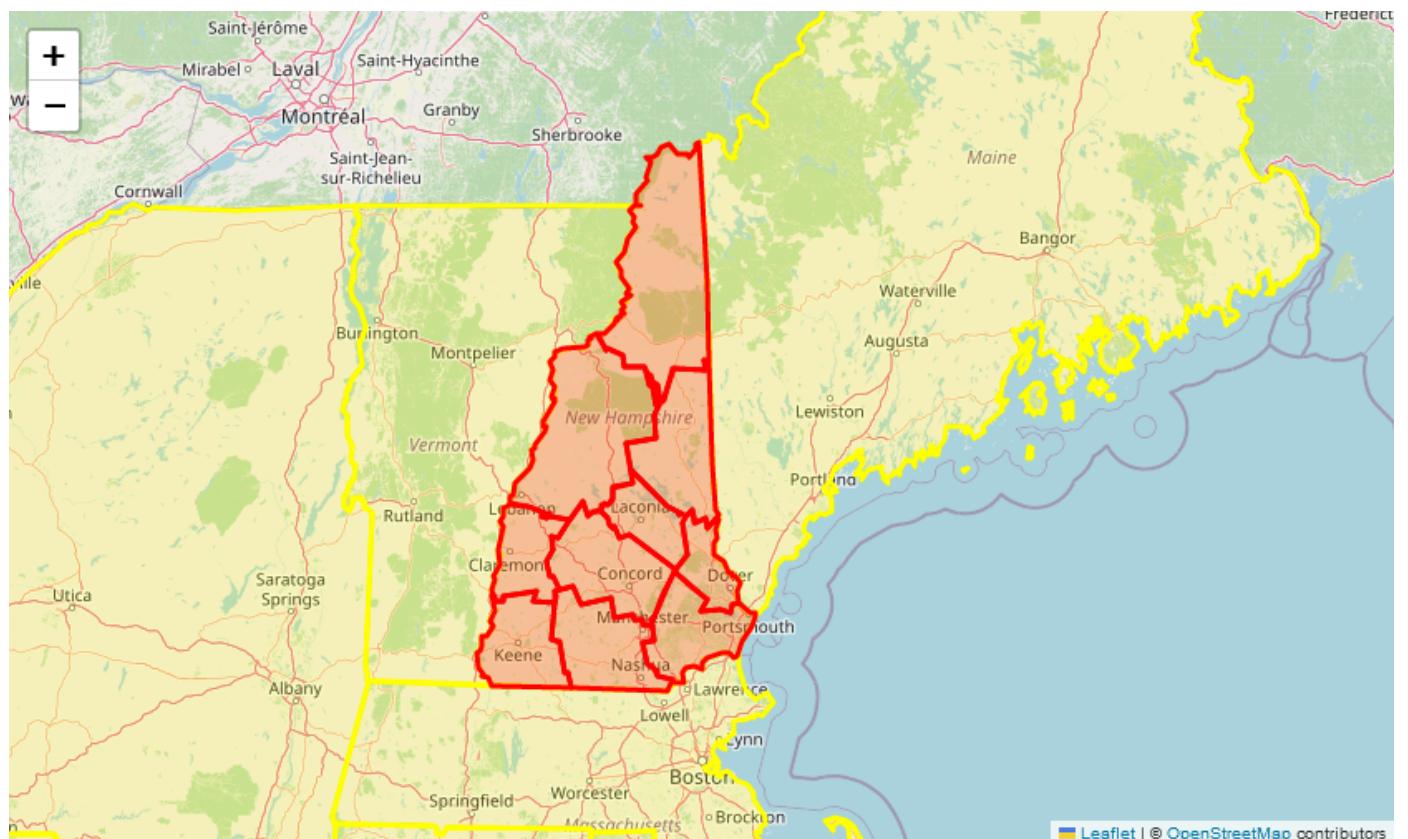
```
SELECT c.county, c.state_abrv, c.geom  
FROM us_counties c, us_states s  
WHERE s.state = 'New Hampshire'  
AND SDO_RELATE ( c.geom,s.geom, 'mask=INSIDE') = 'TRUE';
```

```
SELECT c.county, c.state_abrv, c.geom  
FROM us_counties c, us_states s  
WHERE s.state = 'New Hampshire'  
AND SDO_RELATE ( c.geom,s.geom, 'mask=COVEREDBY') = 'TRUE';
```

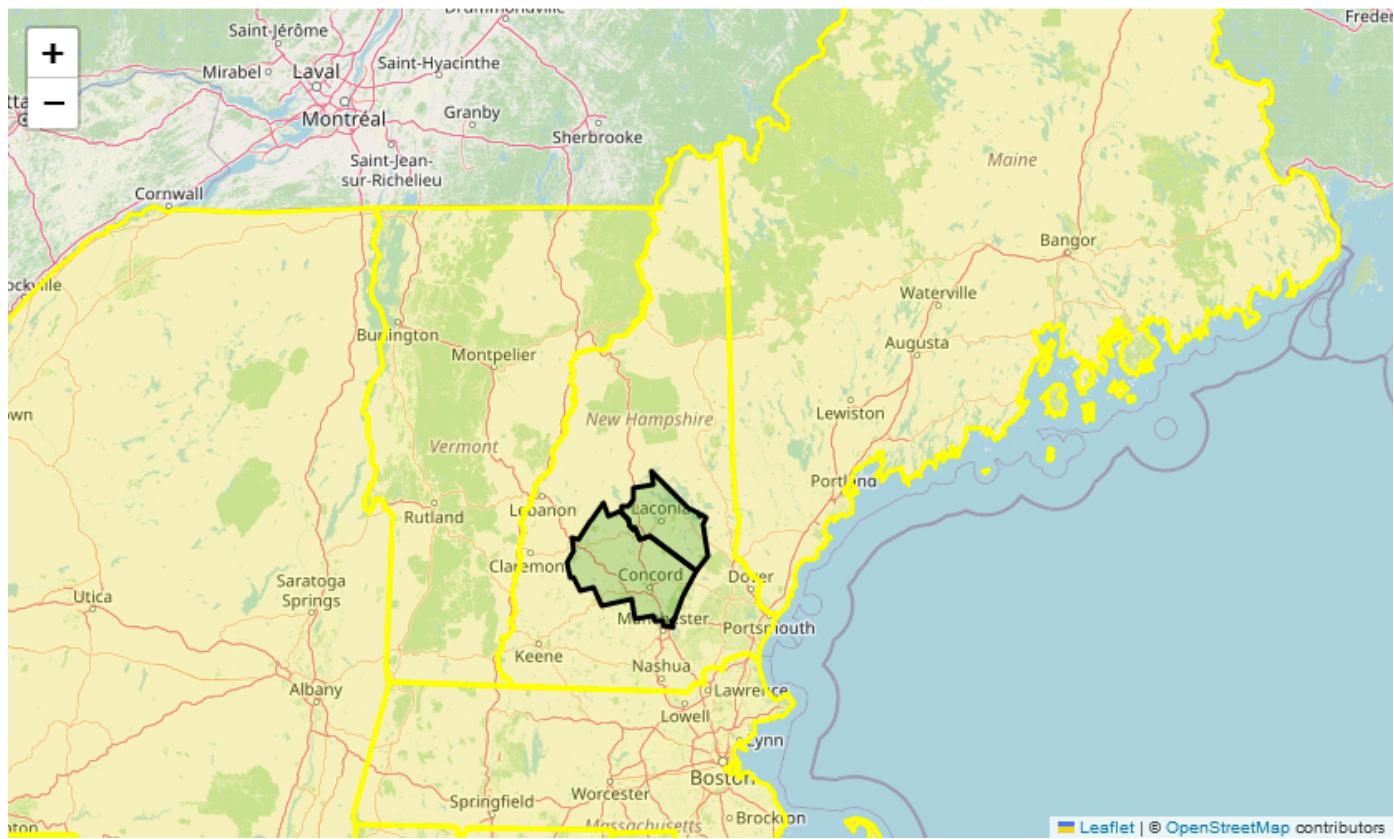
W przypadku wykorzystywania narzędzia SQL Developer, w celu wizualizacji danych na mapie należy użyć podzapytania (podobnie jak w poprzednim zadaniu)

Wyniki, zrzut ekranu, komentarz

INSIDE+COVEREDBY

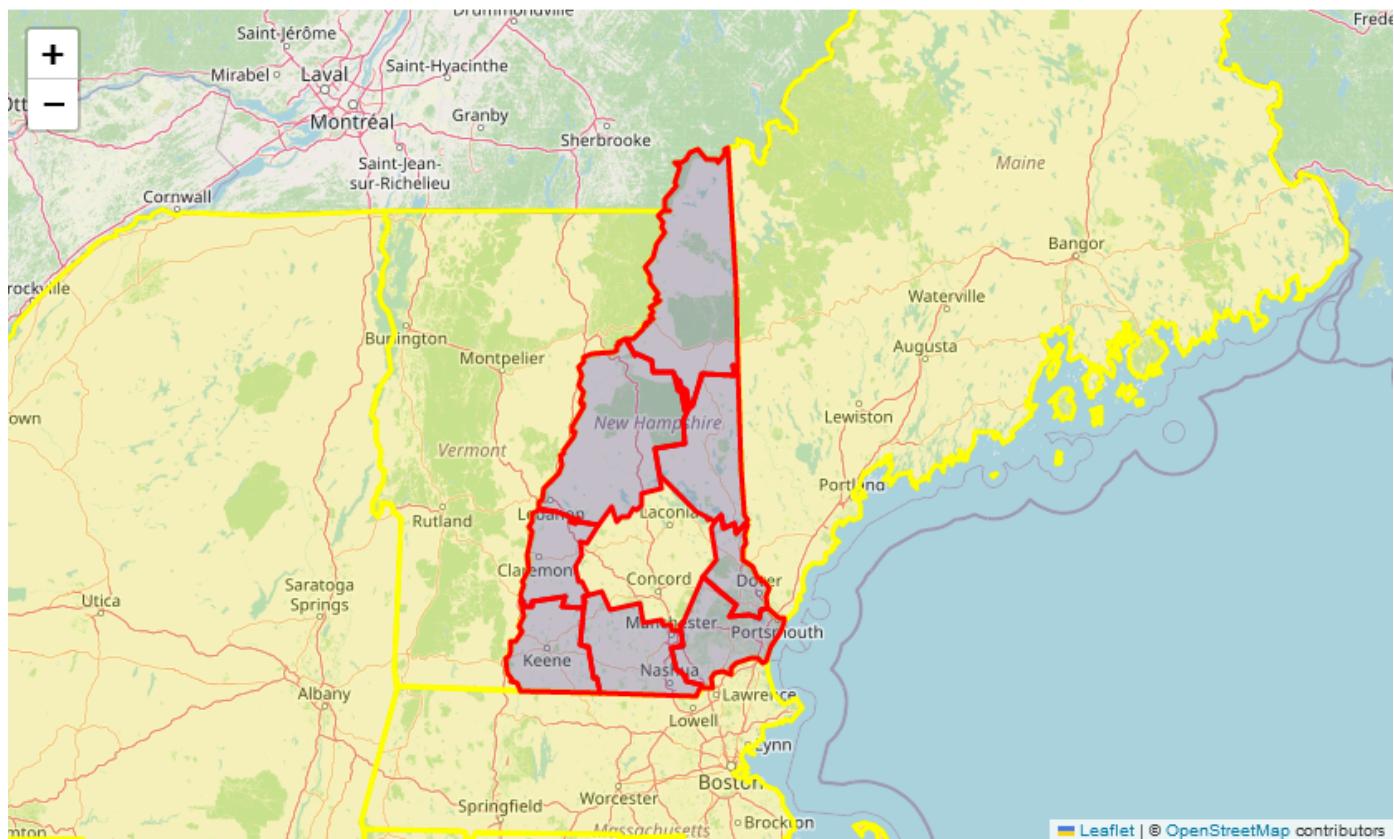


INSIDE



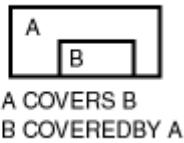
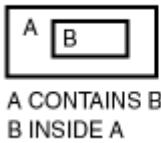
Leaflet | © OpenStreetMap contributors

COVEREDBY



Leaflet | © OpenStreetMap contributors

Wizualizacja masek:



W operatorze SDO_RELATE wiele masek można ze sobą łączyć za pomocą znaku +, który jest odpowiednikiem użycia operacji logicznej OR.

Zadanie 5

Znajdź wszystkie miasta w odległości 50 mili od drogi (us_interstates) I4

Pokaż wyniki na mapie

```
SELECT * FROM us_interstates
WHERE interstate = 'I4'

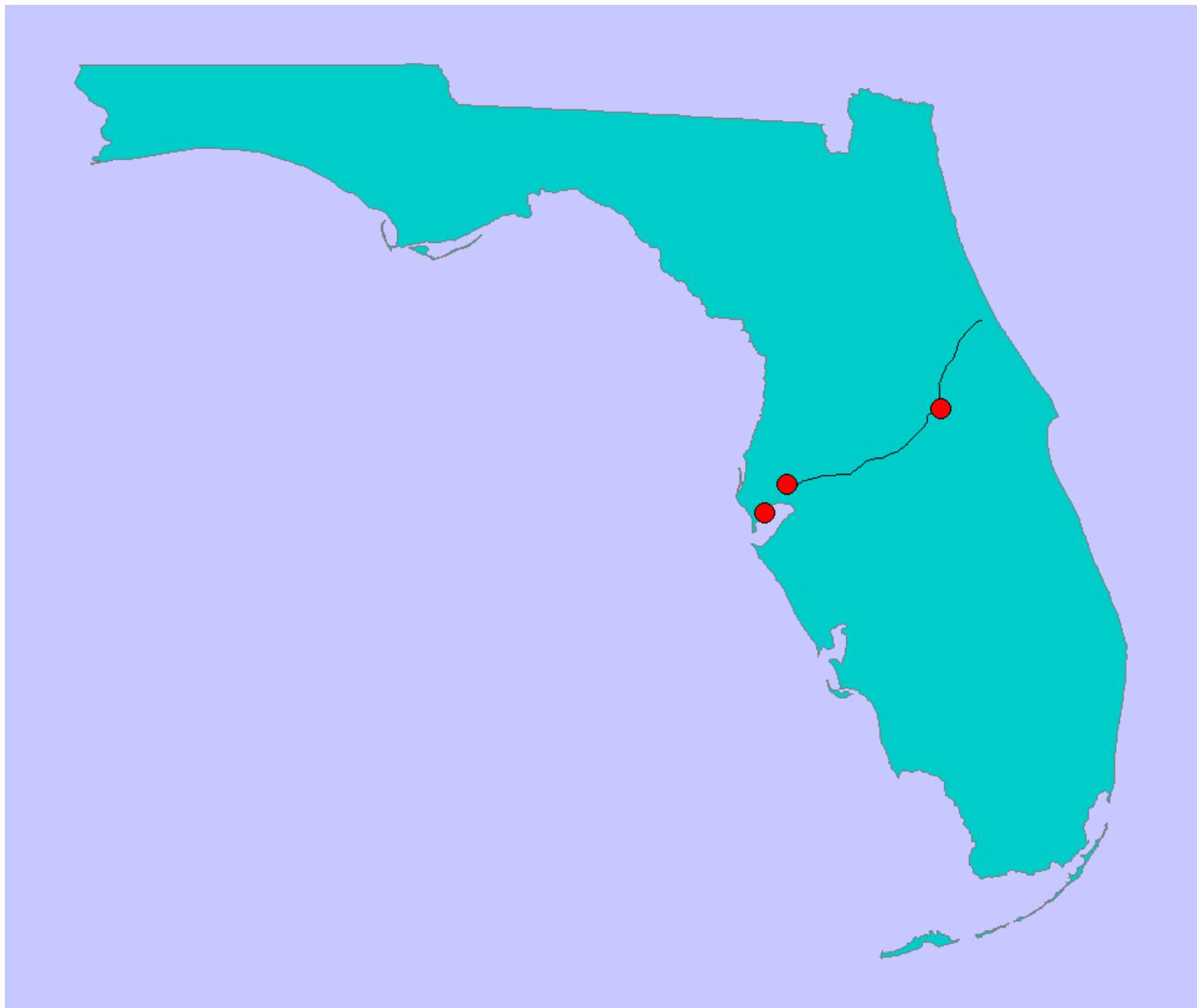
SELECT * FROM us_states
WHERE state_abrv = 'FL'

SELECT c.city, c.state_abrv, c.location
FROM us_cities c
WHERE ROWID IN
(
  SELECT c.rowid
  FROM us_interstates i, us_cities c
  WHERE i.interstate = 'I4'
  AND sdo_within_distance (c.location, i.geom, 'distance=50 unit=mile'
)
```

Wyniki, zrzut ekranu, komentarz

SQL | All Rows Fetched: 3 in 0.061 seconds

	CITY	STATE_ABRV	LOCATION
1	St Petersburg	FL	[MDSYS.SDO_GEOOMETRY]
2	Tampa	FL	[MDSYS.SDO_GEOOMETRY]
3	Orlando	FL	[MDSYS.SDO_GEOOMETRY]



Mamy 3 takie miasta

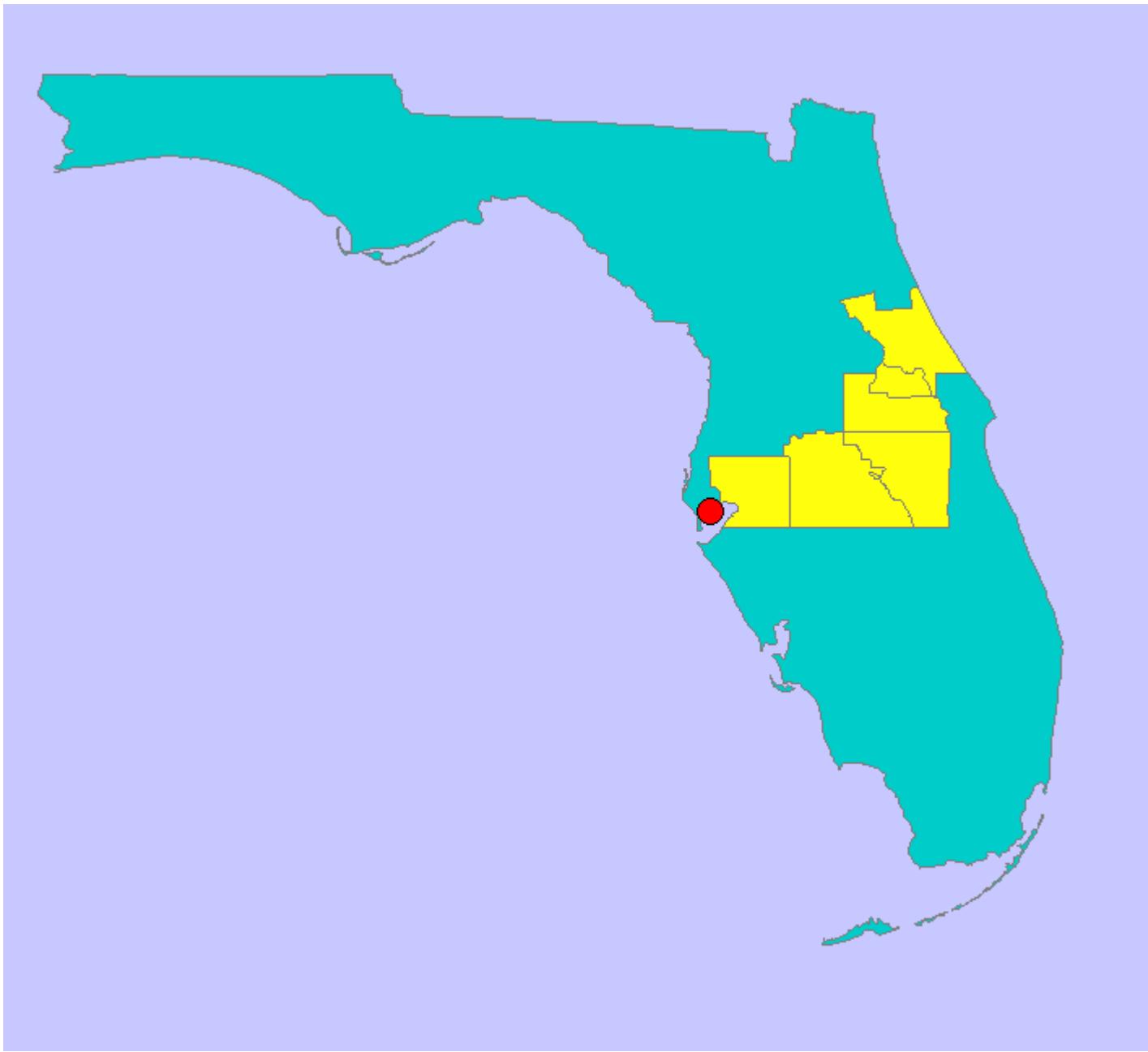
Dodatkowo:

- Znajdz wszystkie jednostki administracyjne przez które przechodzi droga I4

```
SELECT p.id
FROM us_counties p, us_interstates s
WHERE s.interstate = 'I4'
and SDO_ANYINTERACT (p.geom, s.geom) = 'TRUE'
```

ID
1 363
2 367
3 343
4 362
5 373
6 378

```
SELECT pp.county, pp.geom FROM us_counties pp
WHERE id IN
(
  SELECT p.id
  FROM us_counties p, us_interstates s
  WHERE s.interstate = 'I4'
  and SDO_ANYINTERACT (p.geom, s.geom) = 'TRUE'
)
```



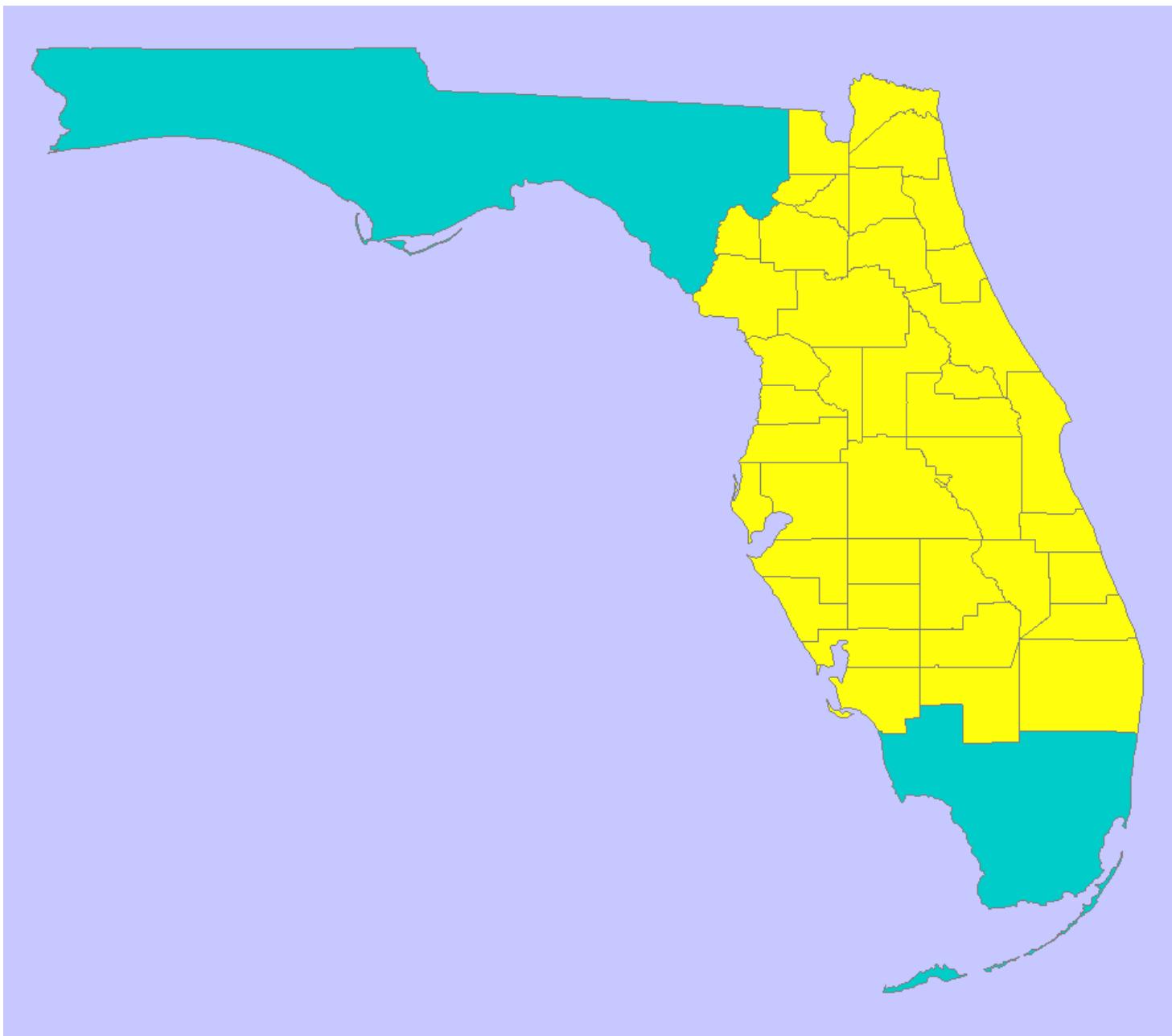
- Mamy 6 takich jednostek, na mapie zostały one oznaczone na żółto
- b) Znajdz wszystkie jednostki administracyjne w pewnej odległości od I4

```
SELECT p.id
FROM us_counties p, us_interstates s
WHERE s.interstate = 'I4'
and SDO_WITHIN_DISTANCE(p.geom, s.geom, 'distance=100 unit=mile') = 'TRUE'
```

    SQL | All Rows Fetched: 40 in 0.103 seconds

ID
1 355
2 372
3 322
4 350
5 339
6 328
7 342
8 336
9 340
10 361
11 357
12 364
13 370
14 345

```
SELECT pp.county, pp.geom FROM us_counties pp
WHERE id IN
(
    SELECT p.id
    FROM us_counties p, us_interstates s
    WHERE s.interstate = 'I4'
    and SDO_WITHIN_DISTANCE(p.geom, s.geom, 'distance=100 unit=mile') = 'TRUE'
)
```



Na mapie na żółto zostały ukazane wszystkie jednostki administracyjne oddalone od I4 o maksymalnie 100 mil. Jest ich 40.

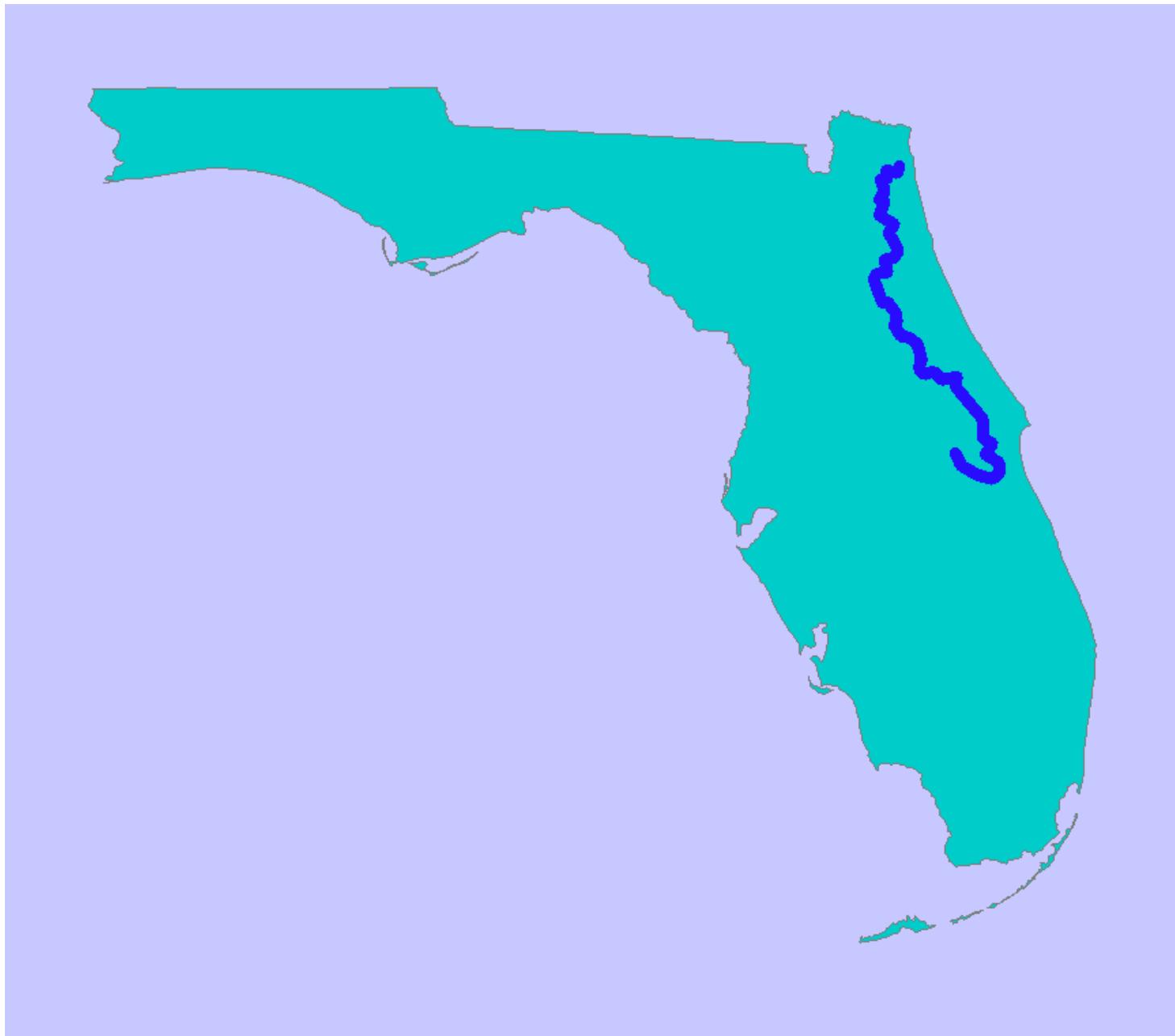
- c) Znajdz rzeki które przecina droga I4

```
SELECT r.id  
FROM us_rivers r, us_interstates s  
WHERE s.interstate = 'I4'  
and SDO_ANYINTERACT (r.geom, s.geom) = 'TRUE'
```

SQL | All Rows Fetched: 1 in 0.086 seconds

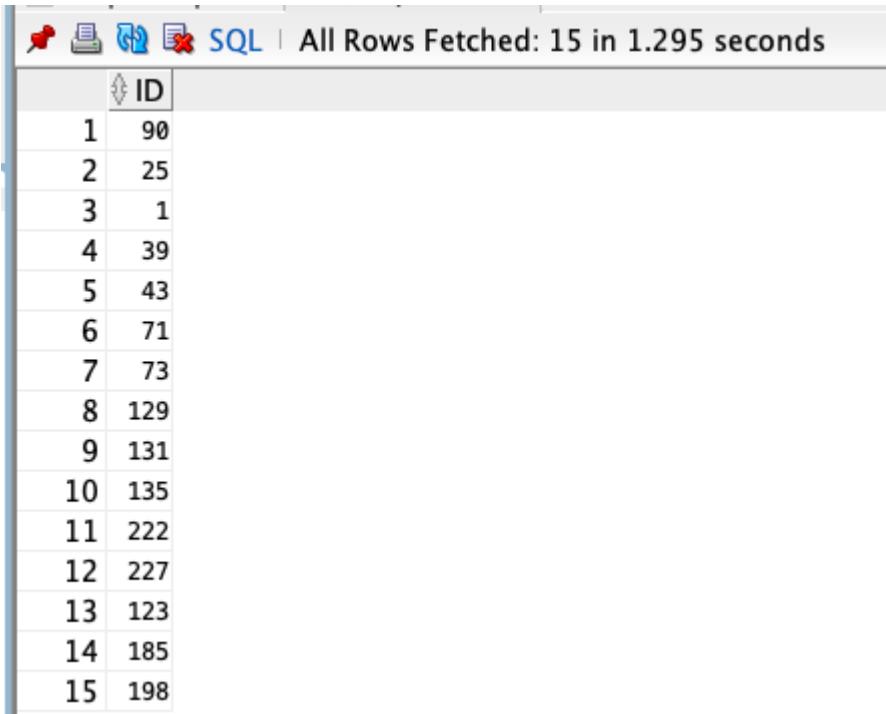
ID
1 43

```
SELECT pp.name, pp.geom FROM us_rivers pp
WHERE id IN
(
    SELECT r.id
    FROM us_rivers r, us_interstates s
    WHERE s.interstate = 'I4'
    and SDO_ANYINTERACT (r.geom, s.geom) = 'TRUE'
)
```



- Istnieje tylko jedna taka rzeka
- d) Znajdz wszystkie drogi które przecinają rzekę Mississippi

```
SELECT s.id
FROM us_interstates s, us_rivers r
WHERE r.name = 'Mississippi'
and SDO_ANYINTERACT (r.geom, s.geom) = 'TRUE'
```

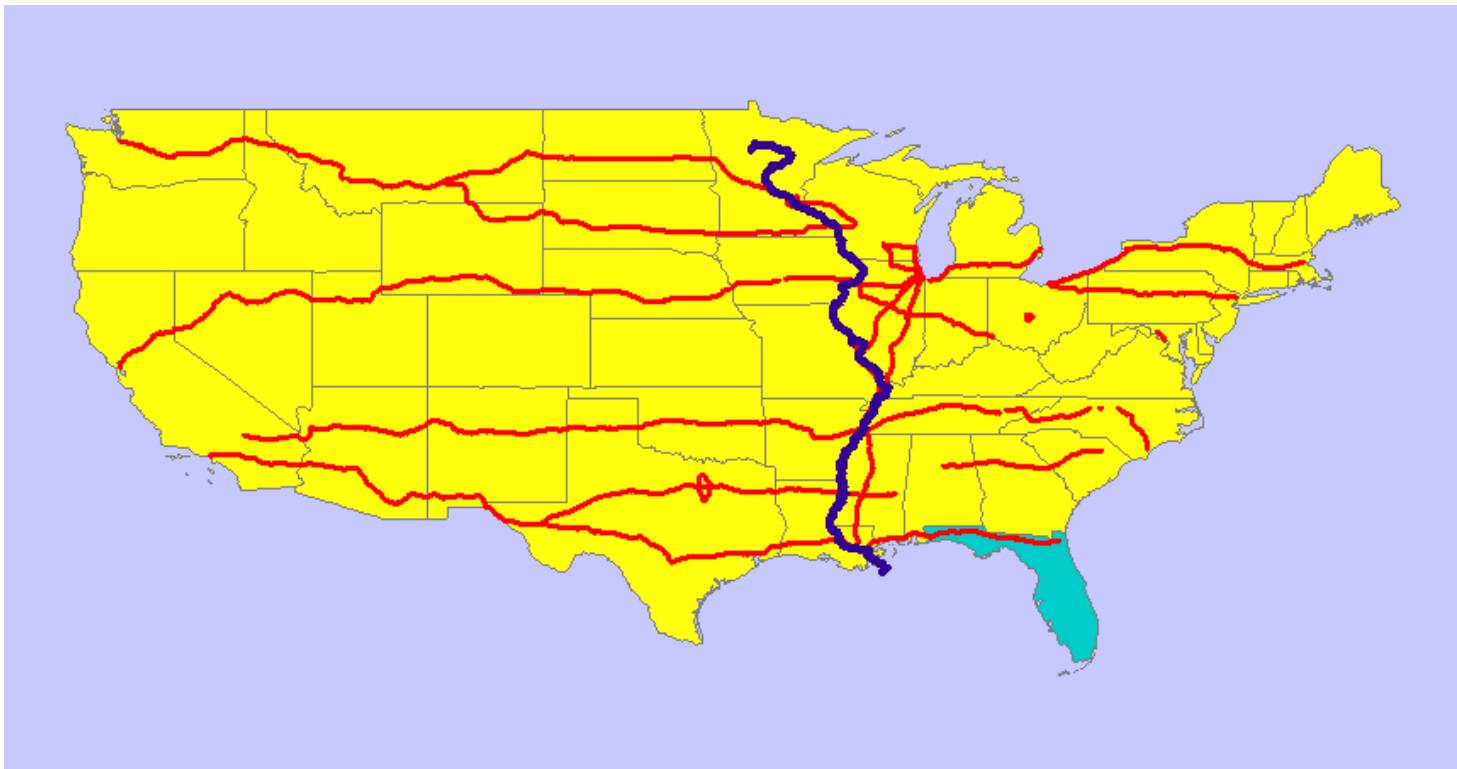


The screenshot shows a database interface with a toolbar at the top featuring icons for search, refresh, and other functions. The main area displays a table with a single column labeled 'ID'. The data consists of 15 rows, each containing a number from 1 to 15. The table has a light gray header row and white background rows.

ID
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```
SELECT pp.interstate, pp.geom FROM us_interstates pp
WHERE id IN
(
    SELECT s.id
    FROM us_interstates s, us_rivers r
    WHERE r.name = 'Mississippi'
    and SDO_ANYINTERACT (r.geom, s.geom) = 'TRUE'
)
SELECT * FROM us_rivers r
WHERE r.name = 'Mississippi'
```

Rzeka missisipi została oznaczona na niebiesko, a drogi na czerwono. Istnieje 15 takich dróg.



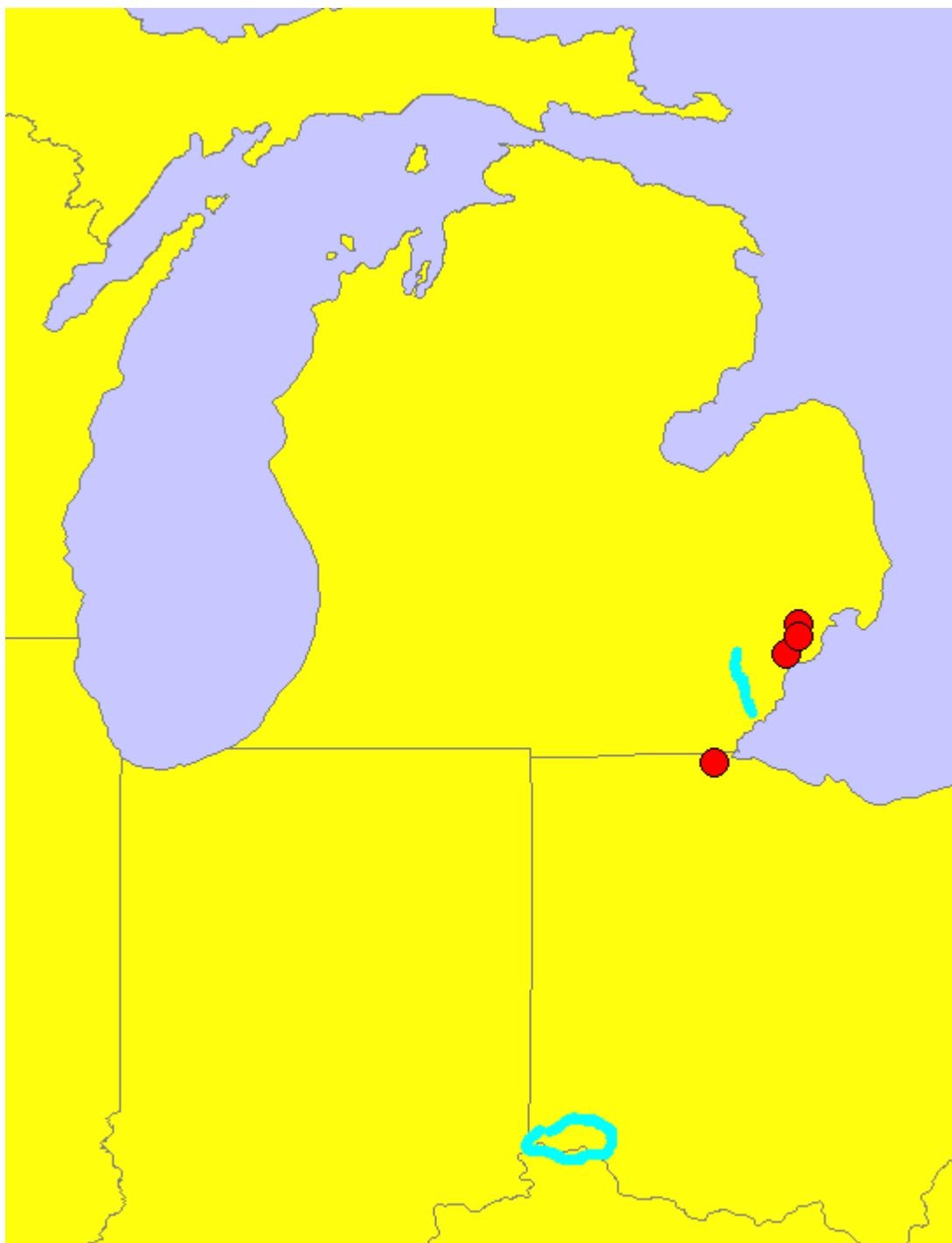
- e) Znajdz wszystkie miasta w odlegości od 15 do 30 mil od drogi 'I275'

```
SELECT c.rowid
FROM us_interstates i, us_cities c
WHERE i.interstate = 'I275'
AND SDO_WITHIN_DISTANCE(c.location, i.geom, 'distance=30 unit=mile') = 'TRUE'
AND NOT SDO_WITHIN_DISTANCE(c.location, i.geom, 'distance=15 unit=mile') = 'TRUE'
```

SQL | All Rows Fetched: 4 in 0.08 seconds

ROWID
1 AABH/ZAAHAADNOHAAw
2 AABH/ZAAHAADNOHAAG
3 AABH/ZAAHAADNOHAB1
4 AABH/ZAAHAADNODAAN

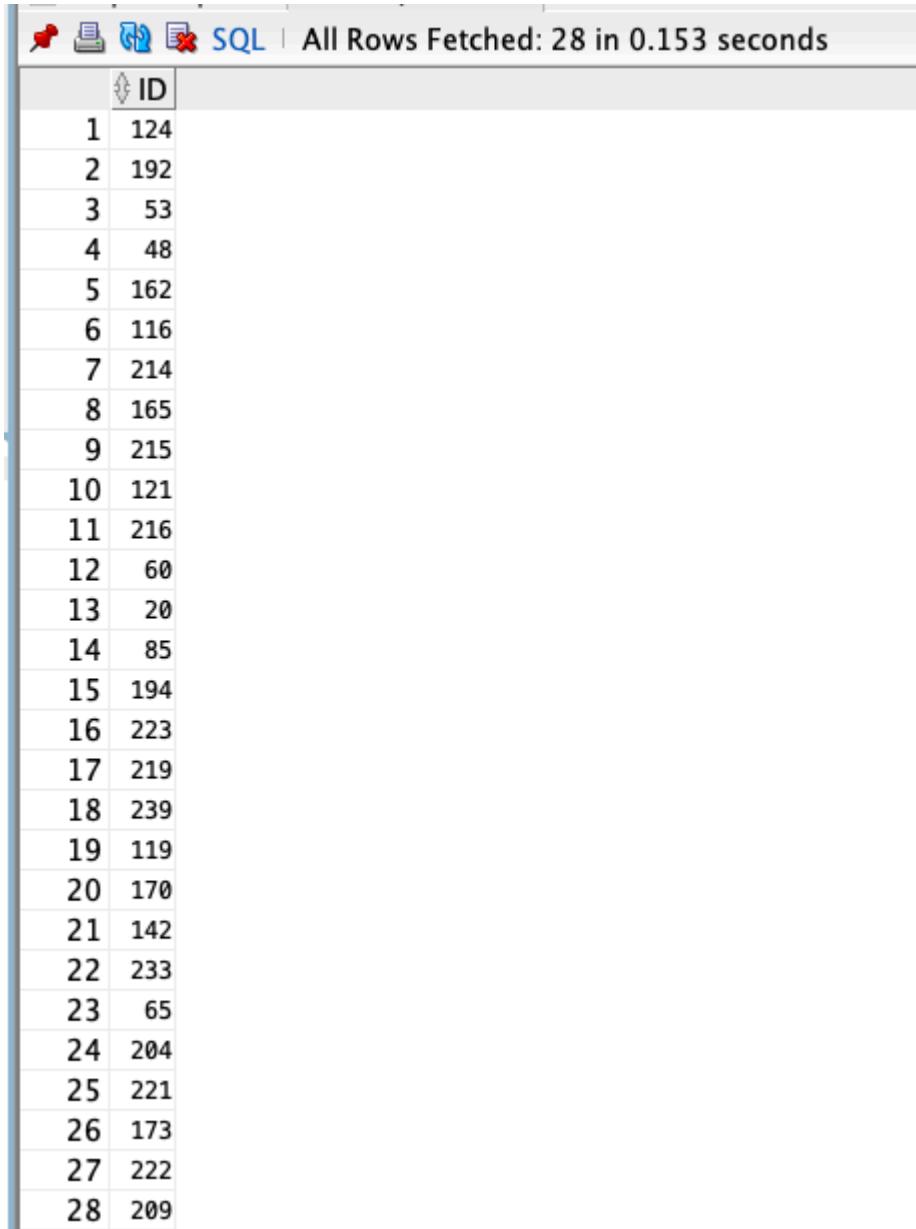
```
SELECT c.city, c.state_abrv, c.location
FROM us_cities c
WHERE ROWID IN
(
  SELECT c.rowid
  FROM us_interstates i, us_cities c
  WHERE i.interstate = 'I275'
  AND SDO_WITHIN_DISTANCE(c.location, i.geom, 'distance=30 unit=mile') = 'TRUE'
  AND NOT SDO_WITHIN_DISTANCE(c.location, i.geom, 'distance=15 unit=mile') = 'TRUE'
)
```



f) Itp. (własne przykłady)

Wszystkie drogi, których jakikolwiek fragment przechodzi przez stan New York

```
SELECT i.id  
FROM us_interstates i, us_states s  
WHERE s.state = 'New York'  
and SDO_ANYINTERACT (i.geom, s.geom) = 'TRUE'
```

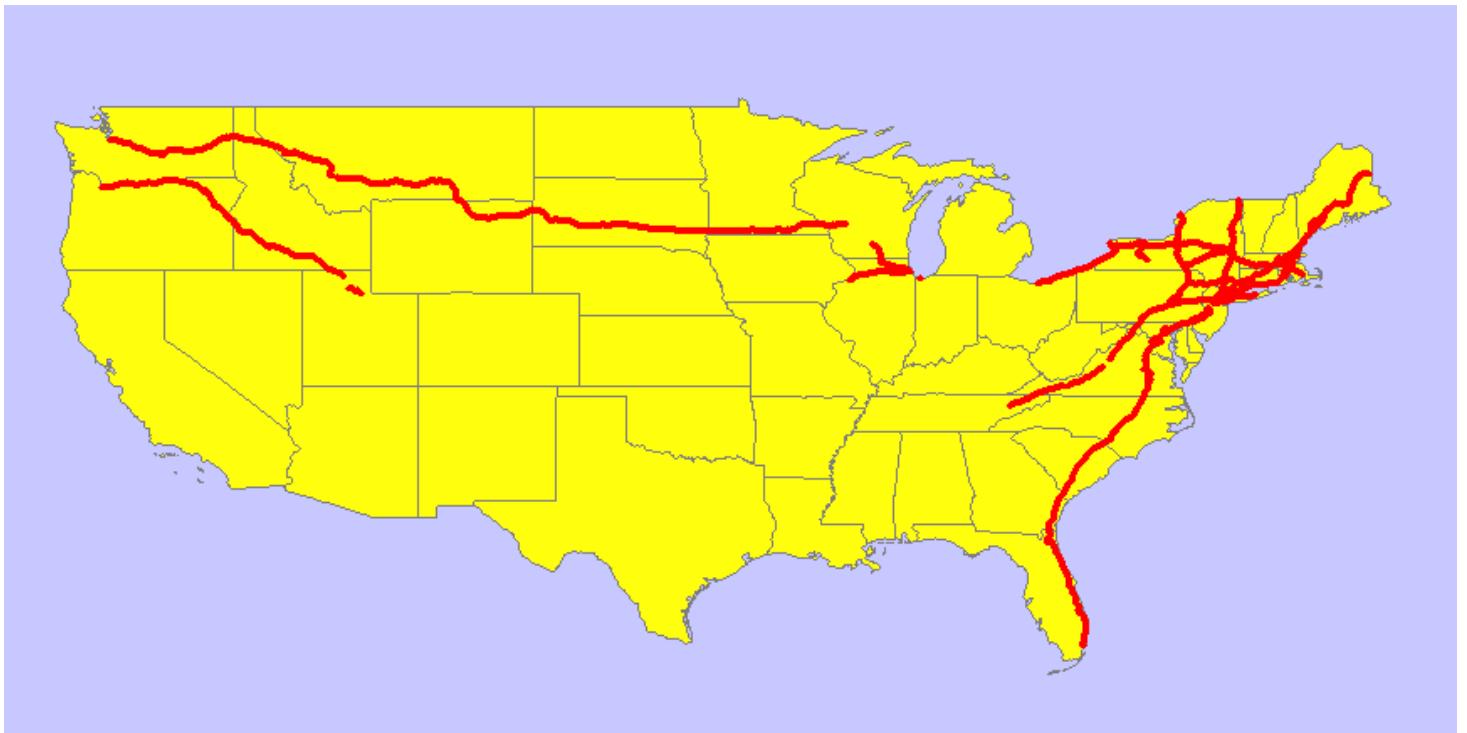


The screenshot shows a database query results window. At the top, there are icons for a key, a document, a magnifying glass, and a red X, followed by the text "SQL" and "All Rows Fetched: 28 in 0.153 seconds". Below this is a table with a single column labeled "ID". The table contains 28 rows of data, each consisting of a number from 1 to 28 followed by a two-digit value.

ID
1 124
2 192
3 53
4 48
5 162
6 116
7 214
8 165
9 215
10 121
11 216
12 60
13 20
14 85
15 194
16 223
17 219
18 239
19 119
20 170
21 142
22 233
23 65
24 204
25 221
26 173
27 222
28 209

```
SELECT pp.interstate, pp.geom FROM us_interstates pp
WHERE id IN
(
    SELECT i.id
    FROM us_interstates i, us_states s
    WHERE s.state = 'New York'
    and SDO_ANYINTERACT (i.geom, s.geom) = 'TRUE'
)
```

Mamy 4 takie miasta.



Wszystkie jednostki administracyjne w odległości maksymalnie 100 mil od rzeki Mississippi

```
SELECT p.id
FROM us_counties p, us_rivers r
WHERE r.name = 'Mississippi'
and SDO_WITHIN_DISTANCE(p.geom, r.geom, 'distance=100 unit=mile') = 'TRUE'
```

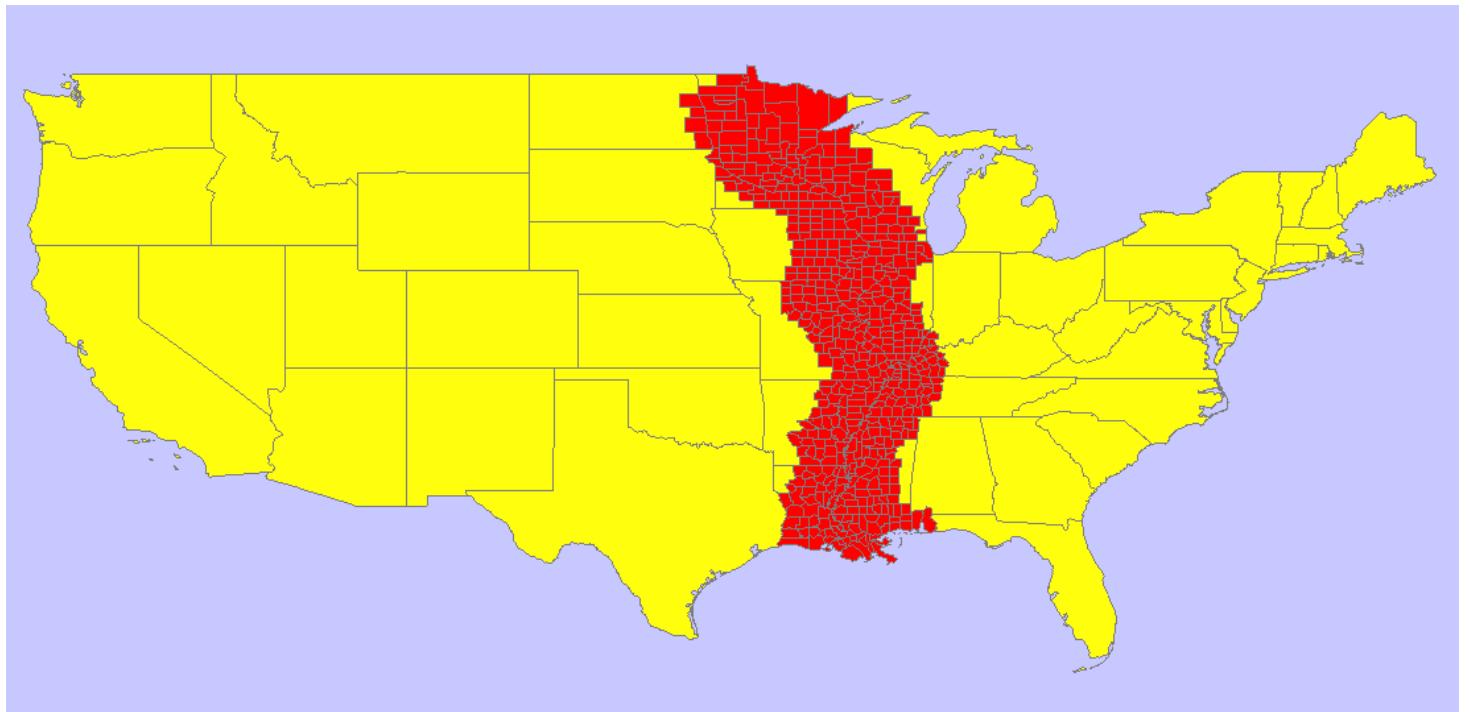
SQL | All Rows Fetched: 568 in 0.811 seconds

ID
540 1323
541 1312
542 1344
543 1347
544 1376
545 1350
546 1395

```

SELECT pp.county, pp.geom FROM us_counties pp
WHERE id IN
(
  SELECT p.id
  FROM us_counties p, us_rivers r
  WHERE r.name = 'Mississippi'
  and SDO_WITHIN_DISTANCE(p.geom, r.geom, 'distance=100 unit=mile') = 'TRUE'
)

```

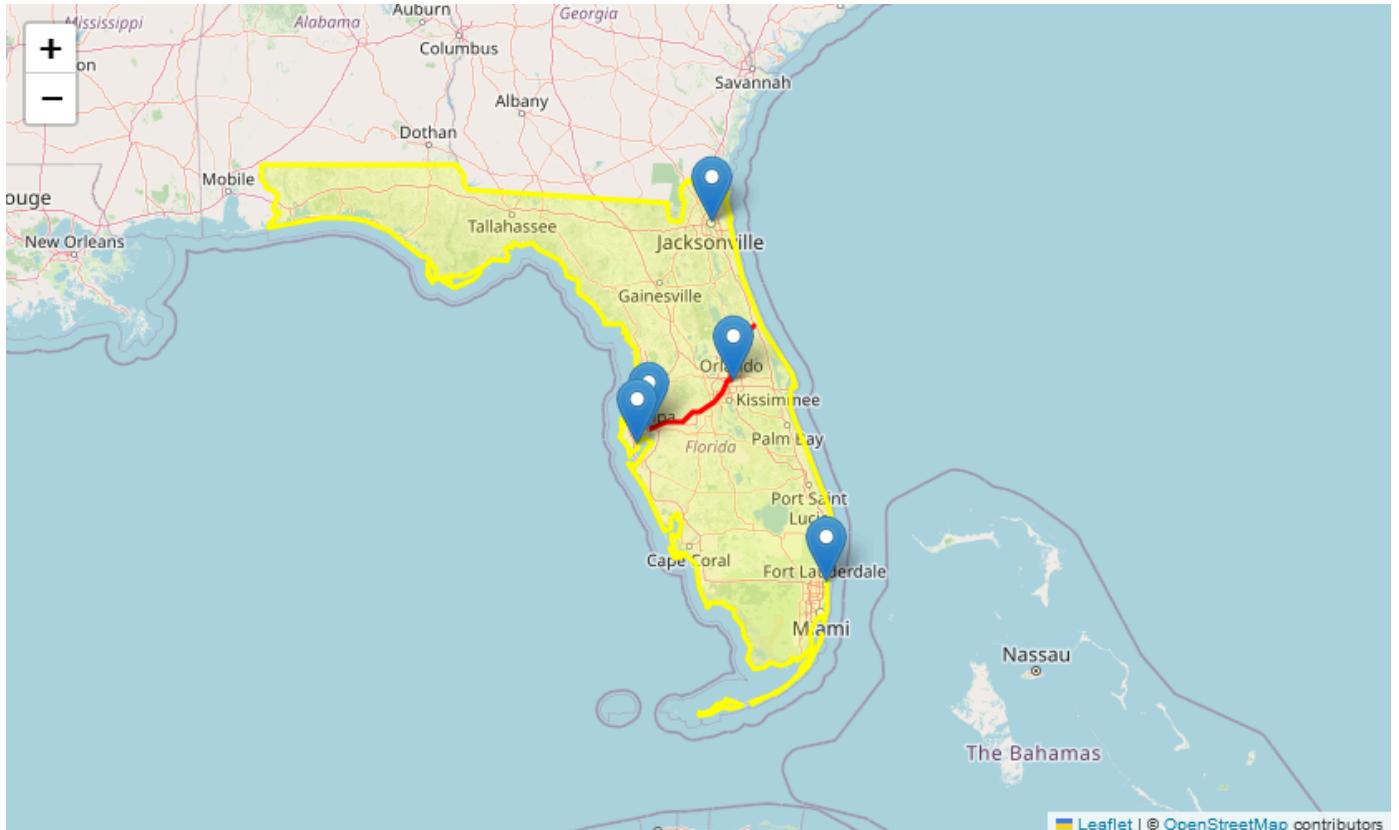


Zadanie 6

Znajdz 5 miast najbliższych drogi I4

```
SELECT c.city, c.state_abrv, c.location
FROM us_interstates i, us_cities c
WHERE i.interstate = 'I4'
AND sdo_nn(c.location, i.geom, 'sdo_num_res=5') = 'TRUE';
```

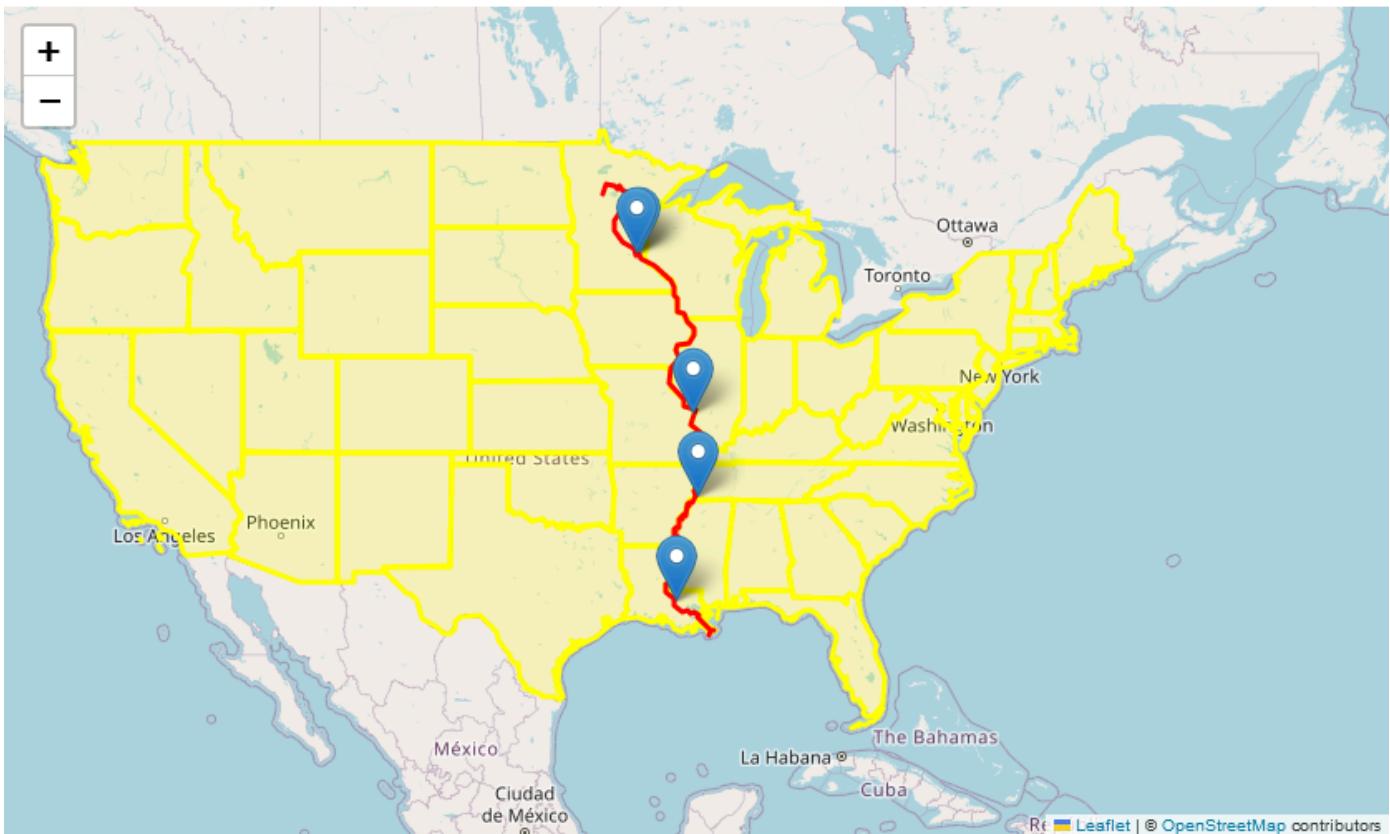
Wyniki, zrzut ekranu, komentarz



```
('Tampa', 'FL')
('Jacksonville', 'FL')
('St Petersburg', 'FL')
('Orlando', 'FL')
('Fort Lauderdale', 'FL')
```

- a) Znajdz kilka miast najbliższych rzece Mississippi

Wyniki, zrzut ekranu, komentarz

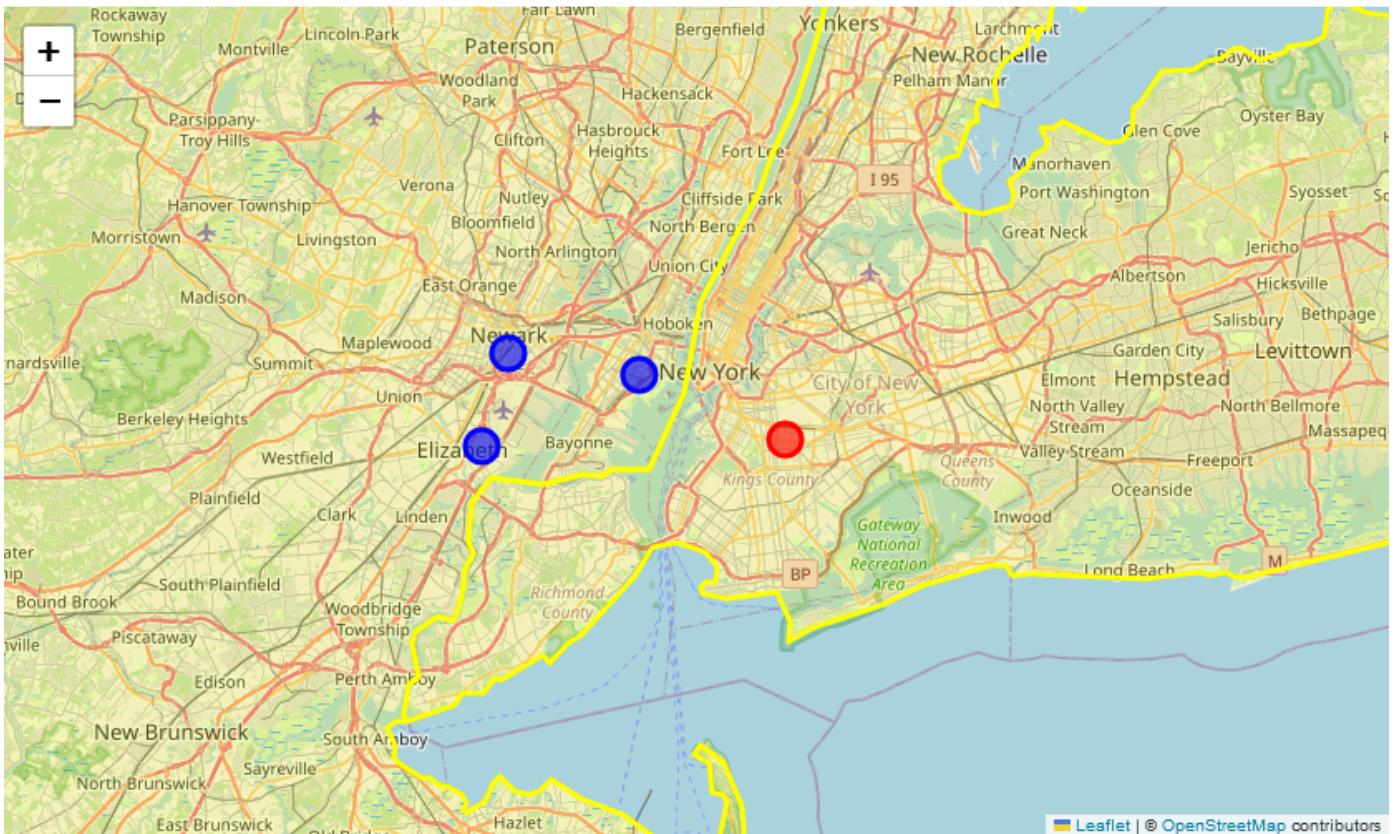


```
('St Paul', 'MN')
('Memphis', 'TN')
('St Louis', 'MO')
('Minneapolis', 'MN')
('Baton Rouge', 'LA')
```

```
SELECT sdo_util.to_wktgeometry(c.location)
FROM us_rivers r, us_cities c
WHERE r.Name = 'Mississippi'
AND sdo_nn(c.location, r.geom, 'sdo_num_res=5') = 'TRUE'
```

- b) Znajdz 3 miasta najbliżej Nowego Jorku

Wyniki, zrzut ekranu, komentarz



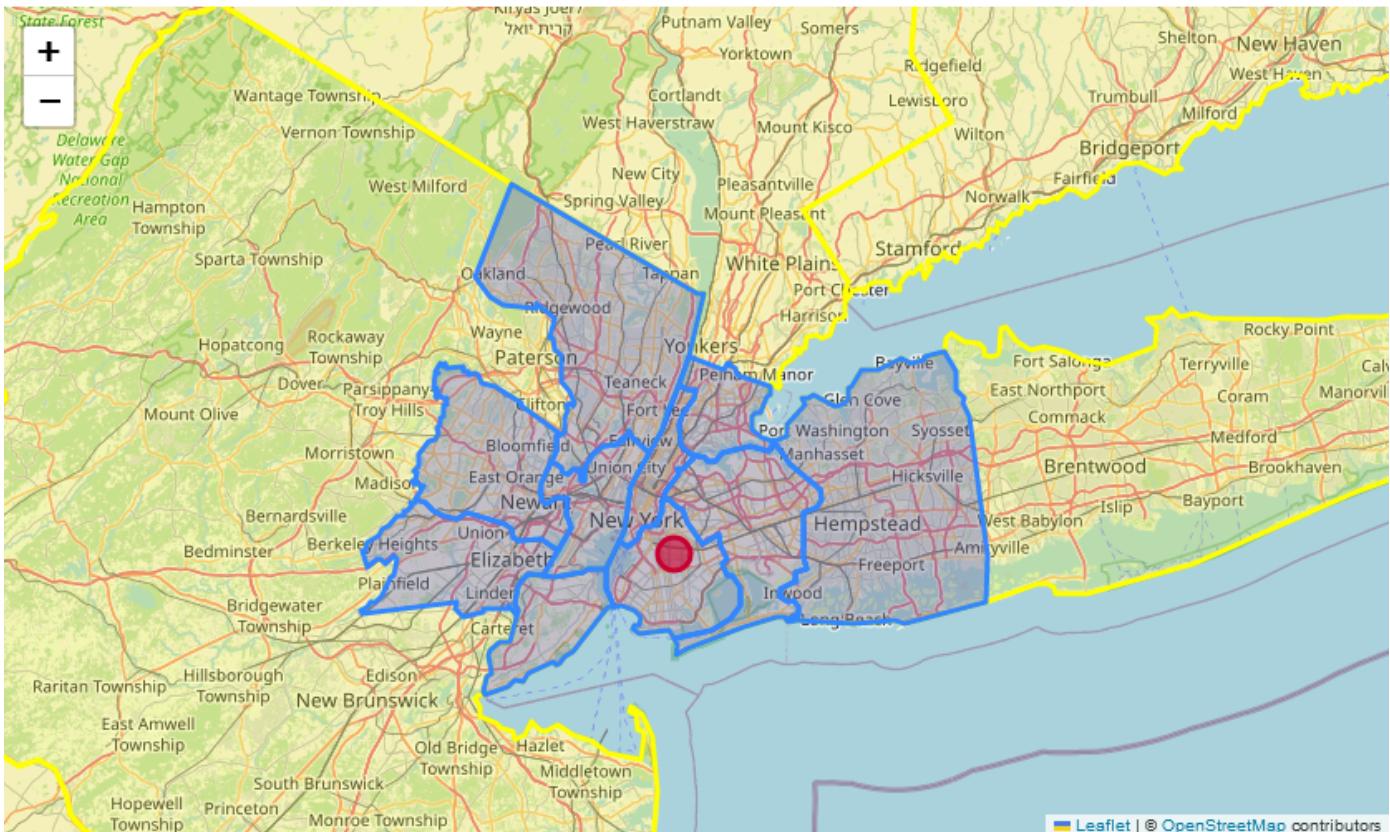
```
('Elizabeth', 'NJ')
('Newark', 'NJ')
('Jersey City', 'NJ')
```

W zapytaniu są 4 najbliższe wyszukania, gdyż jednym z nich będzie Nowy Jork, który następnie zostanie usunięty ze zbioru wynikowego.

```
SELECT sdo_util.to_wktgeometry(c.location) FROM us_cities c
WHERE sdo_nn(
    c.location,
    (SELECT location FROM us_cities WHERE city = 'New York'),
    'sdo_num_res=4'
) = 'TRUE'
AND c.City != 'New York'
```

- c) Znajdz kilka jednostek administracyjnych (us_counties) z których jest najbliżej do Nowego Jorku

Wyniki, zrzut ekranu, komentarz

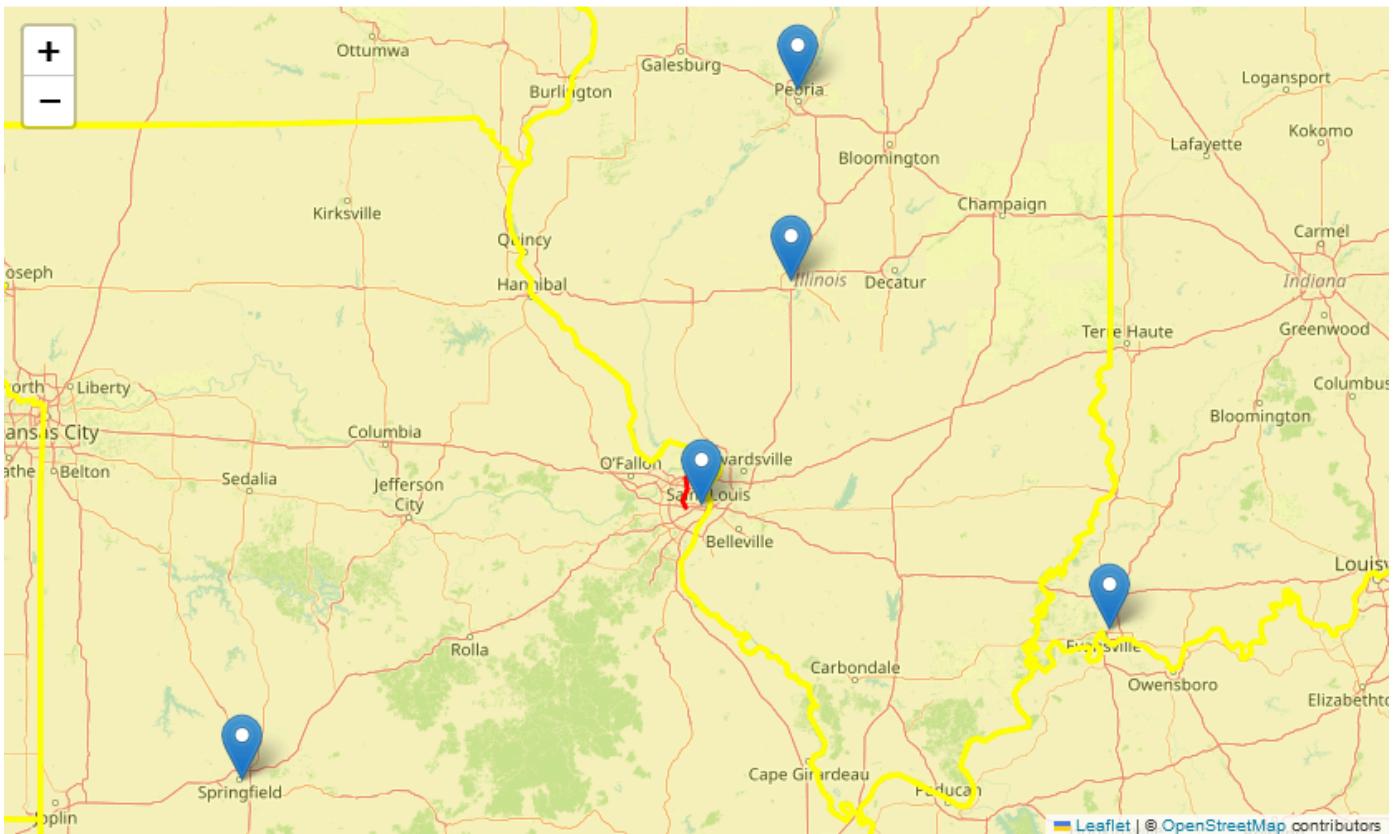


```
('Essex',)
('Hudson',)
('Bronx',)
('Bergen',)
('Queens',)
('Richmond',)
('Kings',)
('Nassau',)
('New York',)
('Union',)
```

```
SELECT sdo_util.to_wktgeometry(co.geom)
FROM us_counties co
WHERE sdo_nn(
    co.geom,
    (SELECT location FROM us_cities WHERE city = 'New York'),
    'sdo_num_res=10'
) = 'TRUE'
```

- d) Znajdz 5 najbliższych miast od drogi 'I170', podaj odległość do tych miast

Wyniki, zrzut ekranu, komentarz



```
('Evansville', 254.637198689794)
('Peoria', 227.686805598134)
('Springfield', 126.815899024404)
('St Louis', 8.63086834124045)
('Springfield', 303.375234373403)
```

```
('Evansville', 254637.198689794)
('Peoria', 227686.805598134)
('Springfield', 126815.899024404)
('St Louis', 8630.86834124044)
('Springfield', 303375.234373403)
```

Zapytanie do wizualizacji

```
SELECT sdo_util.to_wktgeometry(c.location)
FROM us_cities c, us_interstates i
WHERE i.interstate = 'I170'
AND sdo_nn(c.location, i.geom, 'sdo_num_res=5') = 'TRUE'
```

Zapytanie do pomiaru odległości

```
SELECT
    pp.city,
    pp.location,
    SDO_GEOM.SDO_DISTANCE(pp.location, i.geom, 0.005, 'unit=kilometer') AS distance
FROM us_cities pp, us_interstates i
WHERE i.interstate = 'I170'
AND SDO_NN(pp.location, i.geom, 'sdo_num_res=5') = 'TRUE'
```

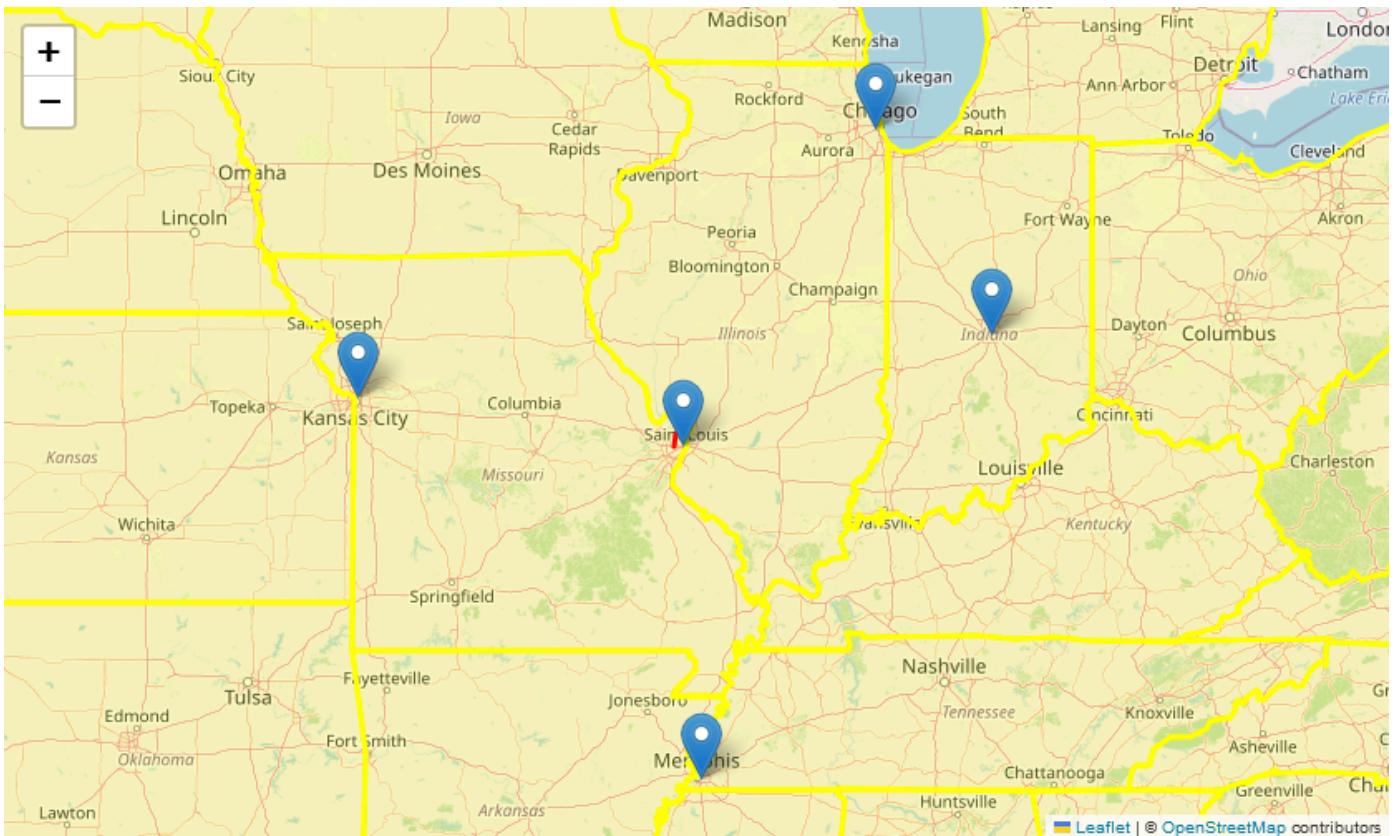
Alternatywne zapytanie do pomiaru odległości

```
SELECT
    pp.city,
    pp.location,
    mdsys.SDO_NN_DISTANCE(1) AS distance
FROM us_cities pp, us_interstates i
WHERE i.interstate = 'I170'
AND SDO_NN(pp.location, i.geom, 'sdo_num_res=5', 1) = 'TRUE'
```

Do pomiaru odległości można użyć operacji SDO_GEOM.SDO_DISTANCE lub mdsys.SDO_NN_DISTANCE

- e) Znajdz 5 najbliższych dużych miast (o populacji powyżej 300 tys) od drogi 'I170'

Wyniki, zrzut ekranu, komentarz



```
('St Louis', 396685, 8.63086834124045)
('Kansas City', 435146, 365.972684073026)
('Indianapolis', 741952, 377.726983800786)
('Memphis', 610337, 393.005139001306)
('Chicago', 2783726, 408.045889800874)
```

Zapytanie do wizualizacji

```
SELECT sdo_util.to_wktgeometry(c.location)
FROM (SELECT *
      FROM us_cities
      WHERE pop90 > 300000
    ) c, us_interstates i
WHERE i.interstate = 'I170'
ORDER BY SDO_GEOM.SDO_DISTANCE(c.location, i.geom, 0.5, 'unit=kilometer')
FETCH FIRST 5 ROWS ONLY
```

Zapytanie do pomiaru odległości

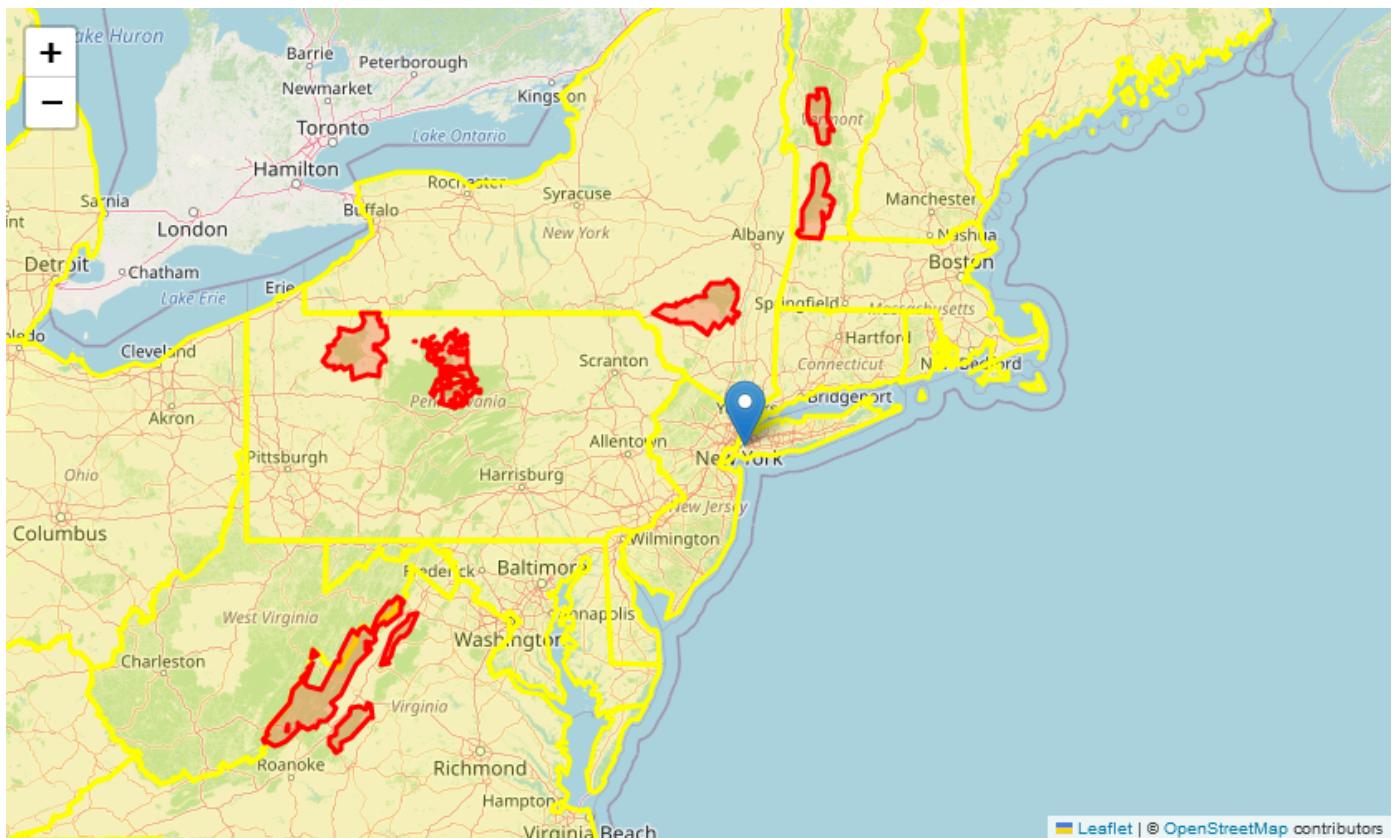
```

SELECT c.city, c.pop90, SDO_GEOGRAPHY.SDO_DISTANCE(c.location, i.geom, 0.5, 'unit=kilometer') as dist
FROM us_cities c, us_interstates i
WHERE i.interstate = 'I170'
AND c.pop90 > 300000
ORDER BY distance
FETCH FIRST 5 ROWS ONLY

```

f) ltp. (własne przykłady)

Parki o powierzchni większej niż 1000 km² oraz w odległości mniejszej niż 500 km od Nowego Jorku

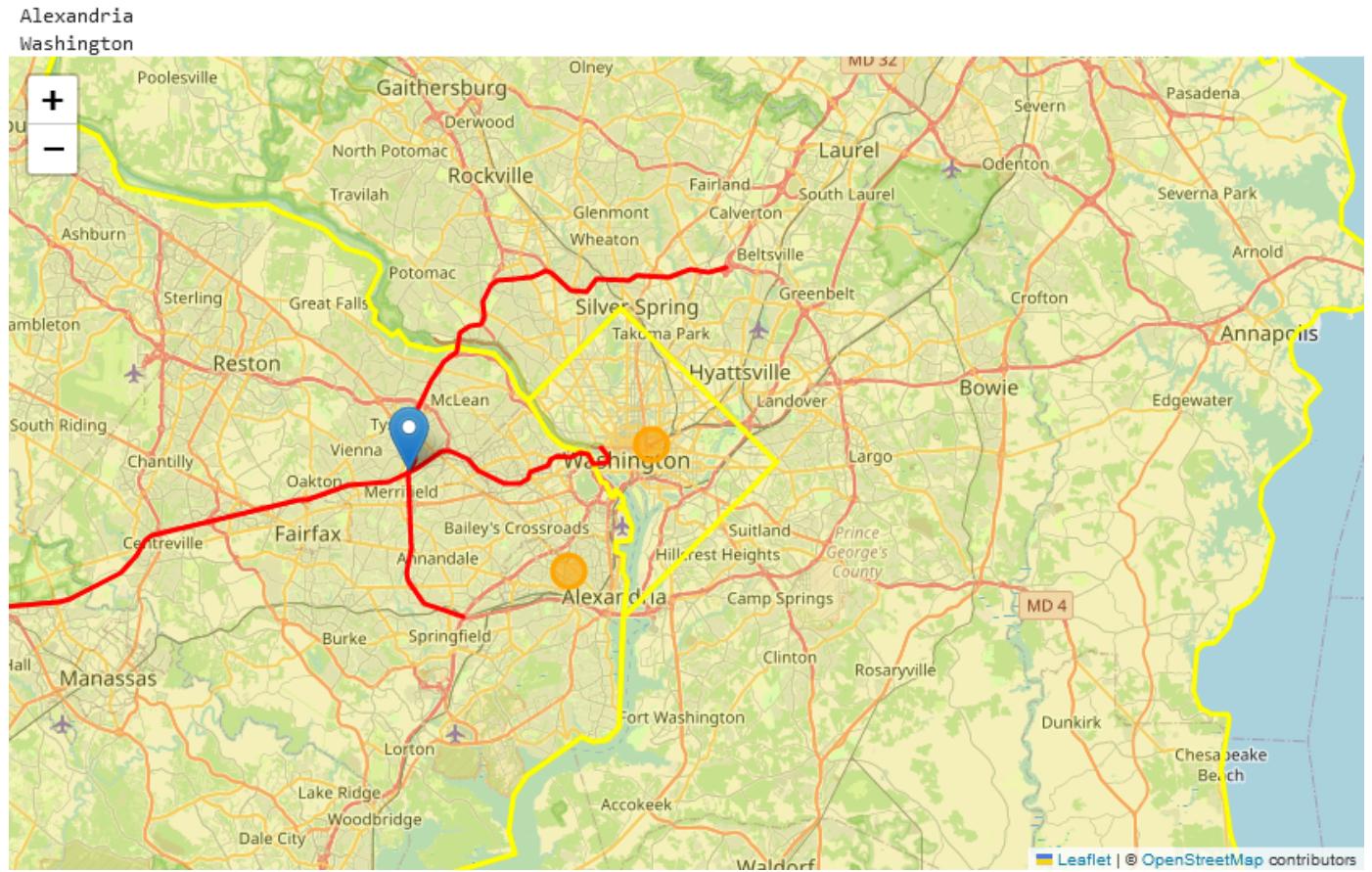


```

SELECT sdo_util.to_wktgeometry(p.geom)
FROM us_parks p
WHERE SDO_GEOGRAPHY.SDO_AREA(p.geom, 0.005) / 1e6 > 1000
AND SDO_WITHIN_DISTANCE(
    p.geom,
    (SELECT location FROM US_CITIES WHERE city = 'New York'),
    'distance=500 unit=km'
) = 'TRUE'

```

Dwa miasta najbliższe przecięciu się dróg I66 oraz I495



```

WITH intersection AS (
  SELECT SDO_GEOM.SDO_INTERSECTION(r1.geom, r2.geom, 0.005) AS geom
  FROM us_interstates r1, us_interstates r2
  WHERE r1.interstate = 'I66'
  AND r2.interstate = 'I495'
  AND SDO_GEOM.SDO_INTERSECTION(r1.geom, r2.geom, 0.005) IS NOT NULL
)
SELECT sdo_util.to_wktgeometry(c.location), c.city
FROM us_cities c
WHERE SDO_NN(c.location, (SELECT geom FROM intersection), 'sdo_num_res=2') = 'TRUE'
  
```

SDO_NN (Nearest Neighbor) jest używany do znajdowania najbliższych obiektów względem określonej geometrii. Umożliwia określenie liczby najbliższych obiektów do zwrócenia (sdo_num_res).

SDO_NN_DISTANCE rozszerza funkcjonalność SDO_NN o dodatkowy atrybut zwracający odległość do najbliższych sąsiadów.

SDO_WITHIN_DISTANCE znajduje wszystkie obiekty, które znajdują się w określonej odległości od zadanej geometrii.

Zadanie 7

Oblicz długość drogi I4

```
SELECT SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer') length
FROM us_interstates
WHERE interstate = 'I4';
```

Wyniki, zrzut ekranu, komentarz

LENGTH
1 212,260756199927

Długość drogi wynosi nieco ponad 212 kilometrów.

Dodatkowo:

- a) Oblicz długość rzeki Mississippi

```
SELECT SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer') length
FROM us_rivers
WHERE name = 'Mississippi';
```

LENGTH
1 3860,32566492228

W poleceniu podmieniamy jedynie tabelę oraz nazwę.

- b) Która droga jest najdłuższa/najkrótsza

Najkrótsza

```
SELECT interstate, SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer') length
FROM us_interstates
ORDER BY length
FETCH NEXT 1 ROW ONLY;
```

The screenshot shows a SQL developer interface with a query window titled "SQL | All Rows Fetched: 1 in 0.24 seconds". The results table has two columns: "INTERSTATE" and "LENGTH". A single row is displayed: Interstate I564 with a length of 0,462140186764249.

	INTERSTATE	LENGTH
1	I564	0,462140186764249

Najdłuższa

```
SELECT interstate, SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer') length
FROM us_interstates
ORDER BY length DESC
FETCH NEXT 1 ROW ONLY;
```

The screenshot shows a SQL developer interface with a query window titled "SQL | All Rows Fetched: 1 in 0.389 seconds". The results table has two columns: "INTERSTATE" and "LENGTH". A single row is displayed: Interstate I90 with a length of 4290,6462617249.

	INTERSTATE	LENGTH
1	I90	4290,6462617249

Wizualizacja

```
SELECT i.interstate, i.geom
FROM us_interstates i
WHERE id IN
(
  SELECT id
  FROM us_interstates
  ORDER BY SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer')
  FETCH NEXT 1 ROW ONLY
)
```

```
SELECT i.interstate, i.geom
FROM us_interstates i
WHERE id IN
(
  SELECT id
  FROM us_interstates
  ORDER BY SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer') DESC
  FETCH NEXT 1 ROW ONLY
)
```



Najdłuższa droga została przedstawiona za pomocą koloru niebieskiego, a najkrótsza za pomocą koloru czerwonego

- c) Która rzeka jest najdłuższa/najkrótsza

Najkrótsza

```
SELECT name, SDO_GEM.SDO_LENGTH (geom, 0.5, 'unit=kilometer') length
FROM us_rivers
ORDER BY length
FETCH NEXT 1 ROW ONLY;
```

SQL | All Rows Fetched: 1 in 0.053 seconds

NAME	LENGTH
Richelieu	1,16169766454518

Najdłuższa

```
SELECT name, SDO_GEM.SDO_LENGTH (geom, 0.5, 'unit=kilometer') length
FROM us_rivers
ORDER BY length DESC
FETCH NEXT 1 ROW ONLY;
```

SQL | All Rows Fetched: 1 in 0.052 seconds

NAME	LENGTH
St. Clair	6950,91937515048

Wizualizacja

```
SELECT r.name, r.geom
FROM us_rivers r
WHERE id IN
(
  SELECT id
  FROM us_rivers
  ORDER BY SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer')
  FETCH NEXT 1 ROW ONLY
)
```

```
SELECT r.name, r.geom
FROM us_rivers r
WHERE id IN
(
  SELECT id
  FROM us_rivers
  ORDER BY SDO_GEOM.SDO_LENGTH (geom, 0.5,'unit=kilometer') DESC
  FETCH NEXT 1 ROW ONLY
)
```



Najdłuższa rzeka przedstawiona jest na czerwono, a najkrótsza na niebiesko.

- d) Które stany mają najdłuższą granicę

```
SELECT state, SDO_GEOM.SDO_LENGTH(geom, 0.5, 'unit=kilometer') AS border_length
FROM us_states
ORDER BY border_length DESC
FETCH FIRST 10 ROWS ONLY
```

STATE	BORDER_LENGTH
1 Alaska	26138,3745019651
2 Texas	6779,84795094551
3 California	4145,76647746918
4 Michigan	4140,12257047995
5 Florida	3725,07858238253
6 Louisiana	3263,313604106
7 Montana	3109,85708647604
8 Minnesota	3001,22942535577
9 North Carolina	2919,74081074775
10 Idaho	2903,44405641629

Za pomocą powyższego polecenia uzyskaliśmy 10 stanów o najdłuższej granicy.

- e) Itp. (własne przykłady)

10 najkrótszych rzek o długości większej niż 1000km

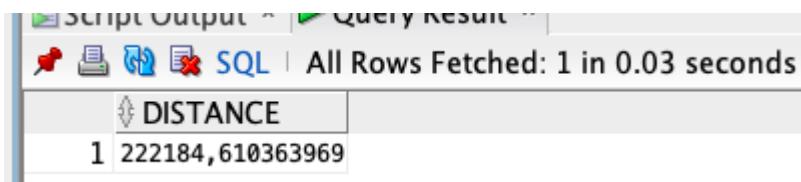
```
SELECT name, SDO_GEOM.SDO_LENGTH(geom, 0.5, 'unit=kilometer') AS length
FROM us_rivers
WHERE SDO_GEOM.SDO_LENGTH(geom, 0.5, 'unit=kilometer') > 1000
ORDER BY length ASC
FETCH FIRST 10 ROWS ONLY
```

NAME	LENGTH
1 Pecos	1049,54072156318
2 Columbia	1056,40746471641
3 St. Lawrence	1122,45505579269
4 Niagara	1312,3788315004
5 Rainy	1328,83766621578
6 Canadian	1335,2221737475
7 Tennessee	1359,90422718882
8 Brazos	1365,37801949504
9 Snake	1388,16254204666
10 Red	1853,53213809113

Oblicz odległość między miastami Buffalo i Syracuse

```
SELECT SDO_GEOM.SDO_DISTANCE ( c1.location, c2.location, 0.5) distance
FROM us_cities c1, us_cities c2
WHERE c1.city = 'Buffalo' and c2.city = 'Syracuse';
```

Wyniki, zrzut ekranu, komentarz



DISTANCE
1 222184,610363969

Jak widać w zapytaniu nie zostało dodane unit=kilometer i dlatego otrzymaliśmy odległość w metrach.

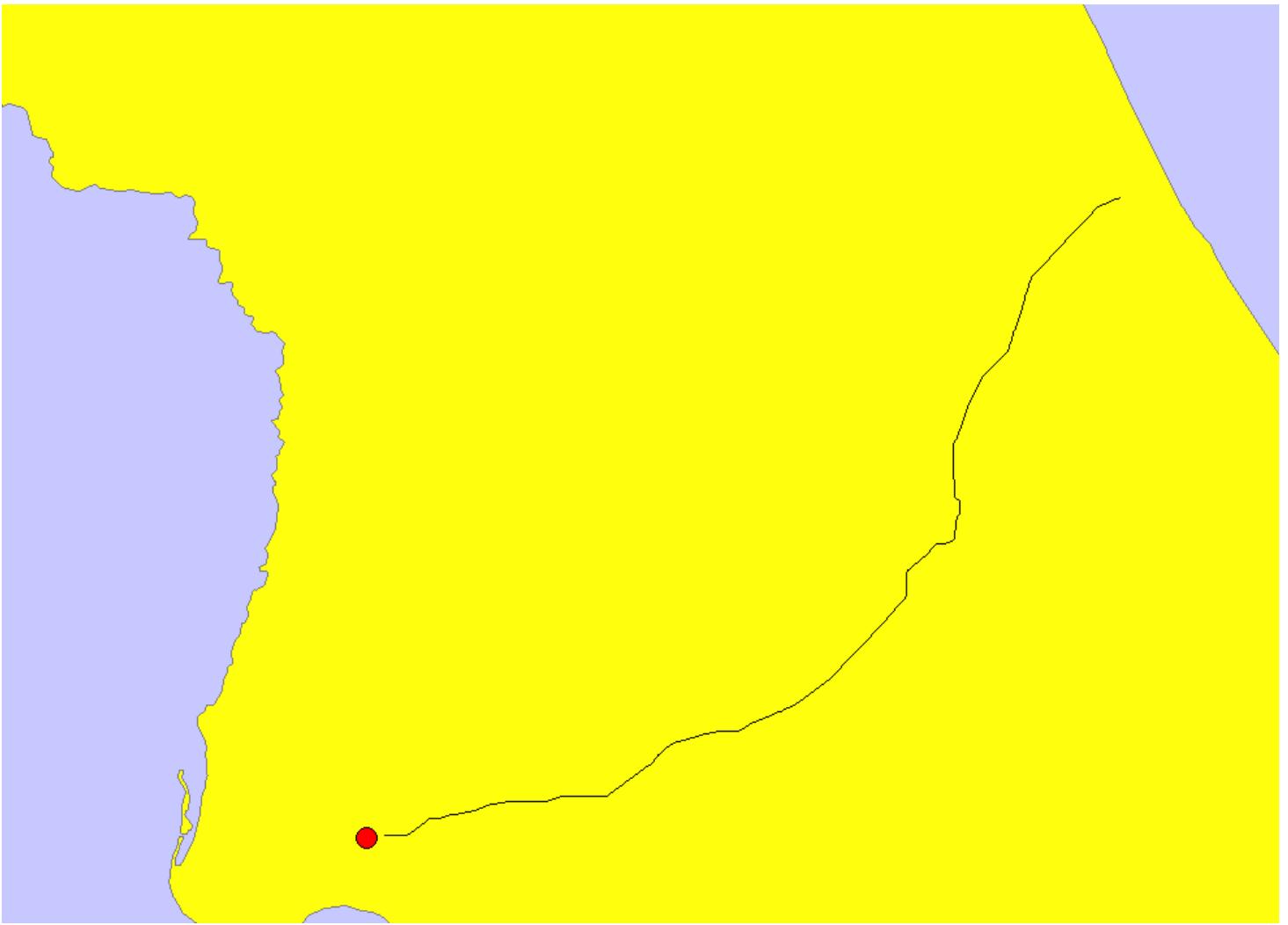
Dodatkowo:

- a) Oblicz odległość między miastem Tampa a drogą I4

```
SELECT SDO_GEOM.SDO_DISTANCE (c1.location, i.geom, 0.5, 'unit=kilometer') distance
FROM us_cities c1, us_interstates i
WHERE c1.city = 'Tampa' and i.interstate = 'I4';
```



DISTANCE
1 3,10391172130556



Odległość to około 3km. Z wizualizacji możemy wywnioskować, że szukana jest odległość pomiędzy dowolnym punktem z autostrady do miasta, które reprezentowane jest przez punkt.

- b) Jaka jest odległość z między stanem Nowy Jork a Florydą

```
SELECT SDO_GEM.SDO_DISTANCE (s1.geom, s2.geom, 0.5, 'unit=kilometer') distance
FROM us_states s1, us_states s2
WHERE s1.state = 'New York' and s2.state = 'Florida';
```

DISTANCE
1 1256,58387785727

Odległość jest mierzona tak jak w poprzednim zapytaniu, pomiędzy dowolnymi dwoma punktami ze stanów.

- c) Jaka jest odległość z między miastem Nowy Jork a Florydą

```

SELECT SDO_GEOM.SDO_DISTANCE (c1.location, s2.geom, 0.5, 'unit=kilometer') distance
FROM us_cities c1, us_states s2
WHERE c1.city = 'New York' and s2.state = 'Florida';

```

SQL All Rows Fetched: 1 in 0.054 seconds	
	DISTANCE
1	1296,59076150732

Różnica wynika z tego, że miasta są reprezentowane jako punkt i mierzymy odległość dokładnie pomiędzy tymi dwoma miastami.

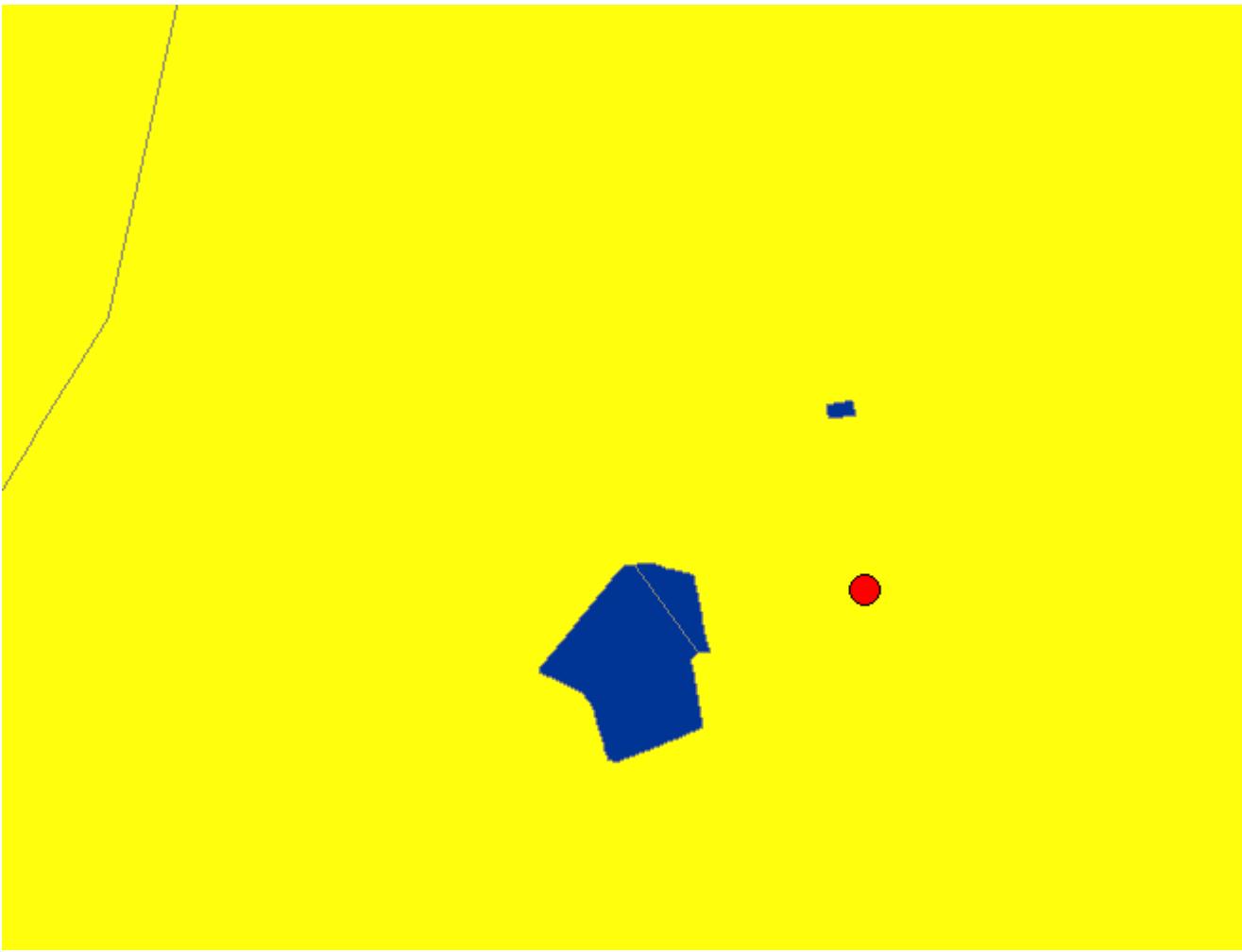
- d) Podaj 3 parki narodowe do których jest najbliżej z Nowego Jorku, oblicz odległości do tych parków

```

SELECT p.name, SDO_GEOM.SDO_DISTANCE(c.location, p.geom, 0.5, 'unit=mile') AS distance
FROM us_parks p, us_cities c
WHERE c.city = 'New York' AND SDO_NN(p.geom, c.location, 'sdo_num_res=3') = 'TRUE'
ORDER BY distance

```

SQL All Rows Fetched: 3 in 0.024 seconds		
	NAME	DISTANCE
1	Institute Park	0,956845723074769
2	Prospect Park	1,06755874464742
3	Thompkins Park	1,32697342712507



Nowy Jork na mapie został oznaczony na czerwono, a parki na niebiesko.

- e) Przetestuj działanie funkcji
 - a. sdo_intersection, sdo_union, sdo_difference
 - b. sdo_buffer
 - c. sdo_centroid, sdo_mbr, sdo_convexhull, sdo_simplify
- f) Itp. (własne przykłady)

Wyniki, zrzut ekranu, komentarz
(dla każdego z podpunktów)

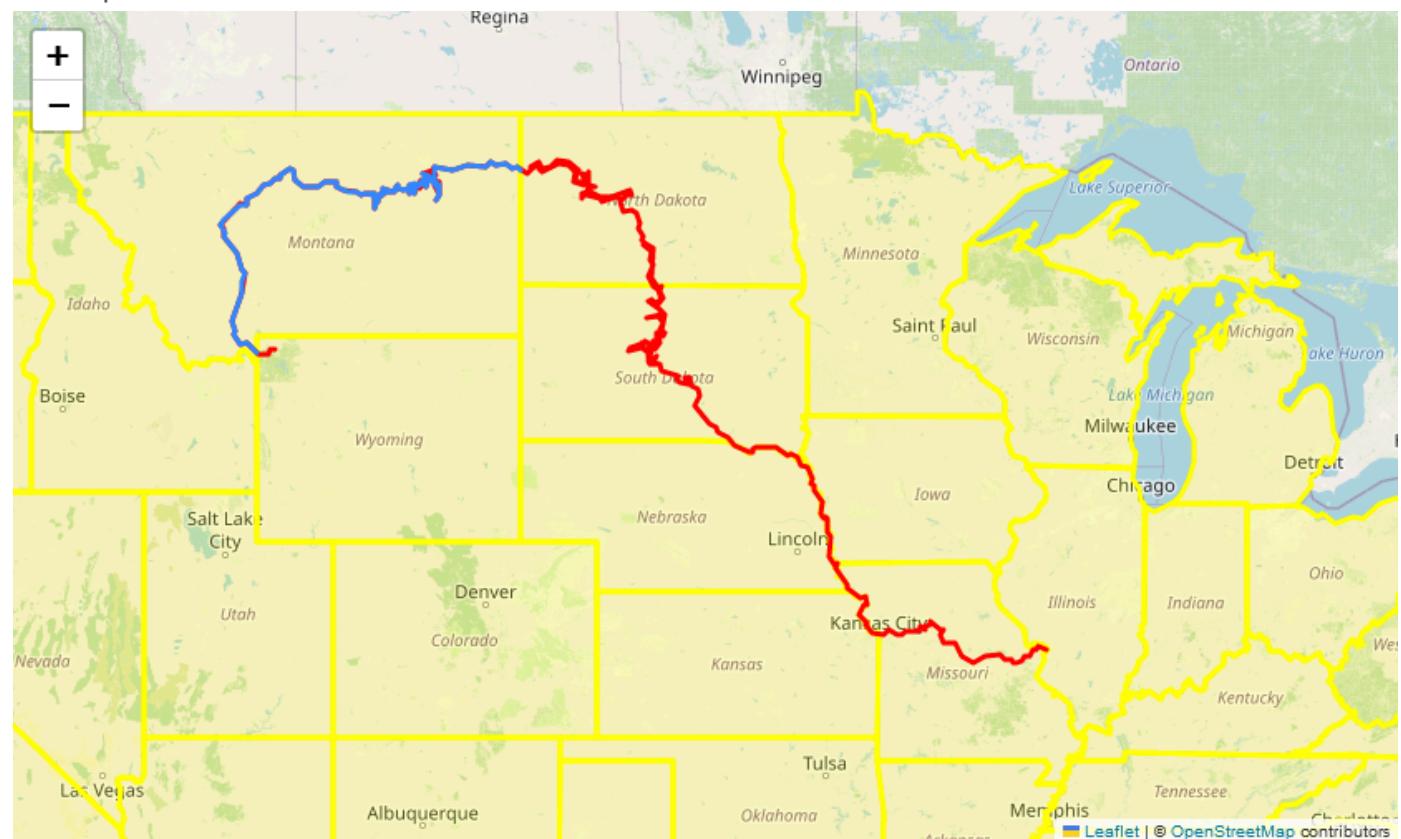
-- ...

Zadanie 8

Wykonaj kilka własnych przykładów/analiz

Wyniki, zrzut ekranu, komentarz

Wyznaczenie w którym stanie rzeka Missouri ma największą długość i zaznaczenie tego odcinka na mapie.

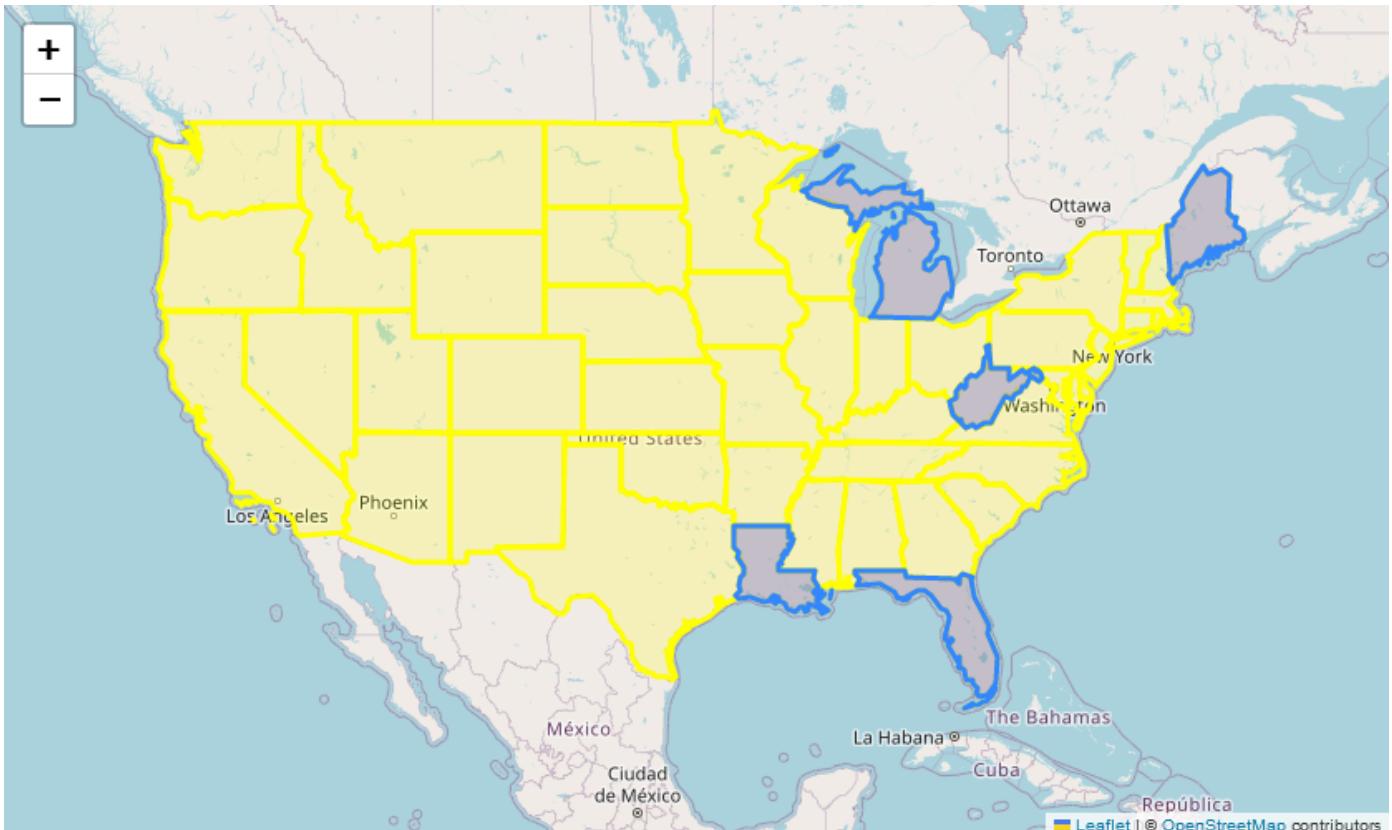


```

WITH river_geom AS (
  SELECT geom
  FROM us_rivers
  WHERE Name = 'Missouri'
),
intersection AS (
  SELECT s.state,
    SDO_GEOGRAPHY.SDO_INTERSECTION(s.geom, r.geom, 0.005) AS common
  FROM us_states s, river_geom r
  WHERE SDO_GEOGRAPHY.SDO_INTERSECTION(s.geom, r.geom, 0.005) IS NOT NULL
),
lengths AS (
  SELECT state,
    SDO_GEOGRAPHY.SDO_LENGTH(common, 0.005) AS length,
    common
  FROM intersection
)
SELECT state, sdo_util.to_wktgeometry(common)
FROM lengths
ORDER BY length DESC
FETCH FIRST 1 ROWS ONLY

```

Wyznaczenie które stany, większe niż 50000 km², mają największy stosunek długości granicy do powierzchni i zaznaczenie tych stanów na mapie.

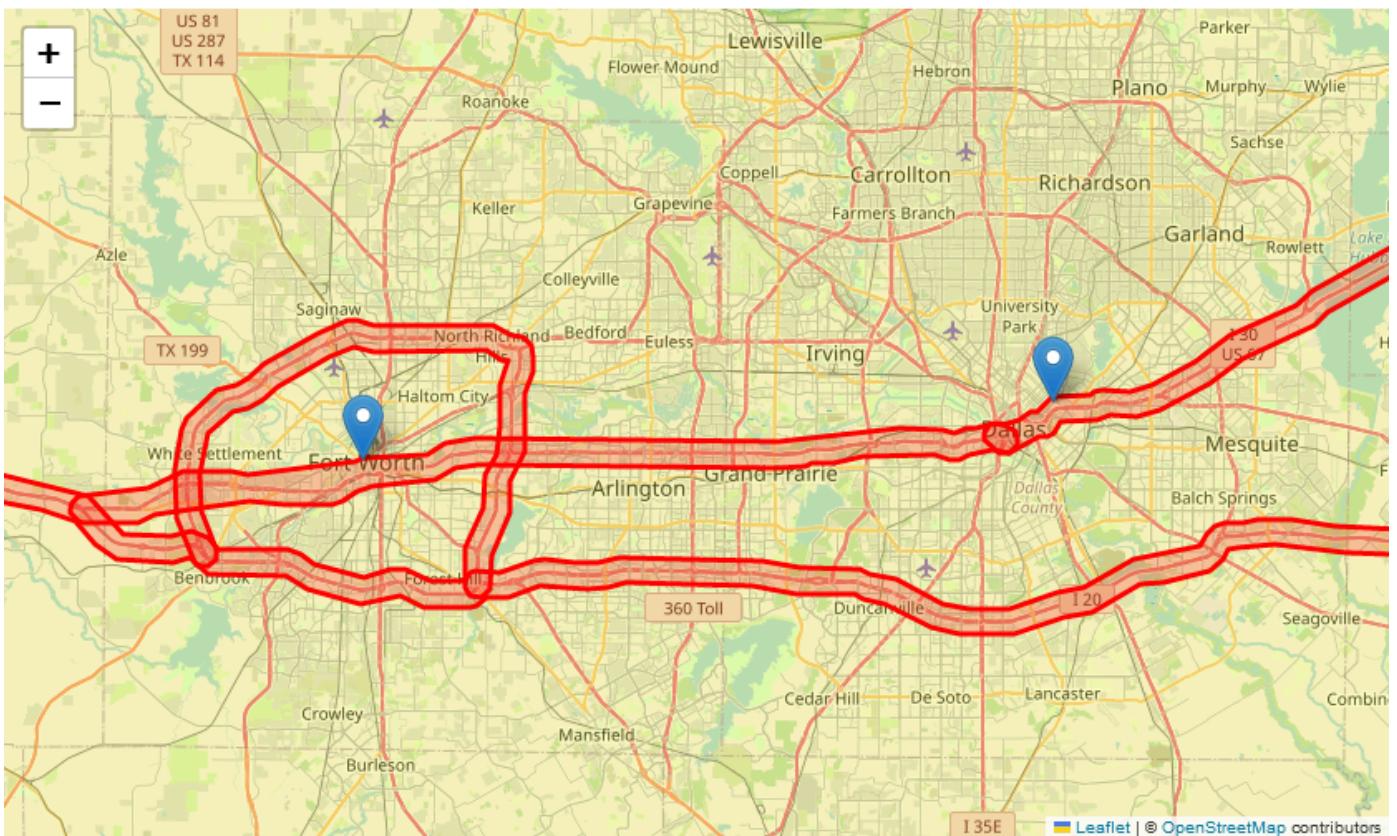


```

WITH state_border_info AS (
    SELECT state,
        SDO_GEOGRAPHY.SDO_LENGTH(geom, 0.005) AS border_length,
        SDO_GEOGRAPHY.SDO_AREA(geom, 0.005) AS area,
        geom
    FROM us_states
    WHERE SDO_GEOGRAPHY.SDO_AREA(geom, 0.005, 'unit=SQ_KM') >= 50000
)
SELECT state, border_length / area AS border_to_area_ratio, sdo_util.to_wktgeometry(geom)
FROM state_border_info
ORDER BY border_to_area_ratio DESC
FETCH FIRST 5 ROWS ONLY

```

Zapytanie o miasta, których centrum leży mniej niż kilometr od drogi międzystanowej



Zapytanie o miasta:

```

WITH t AS (
    SELECT DISTINCT c.city AS city
    FROM us_cities c
    JOIN us_interstates r ON SDO_WITHIN_DISTANCE(c.location, r.geom, 'distance=1 unit=kilometer')
)
SELECT c.city, sdo_util.to_wktgeometry(c.location)
FROM us_cities c
JOIN t ON c.city = t.city

```

Wizualizacja:

```

SELECT sdo_util.to_wktgeometry(SDO_GEM.SDO_BUFFER(geom, 1, 0.005, 'unit=kilometer'))
FROM us_interstates

```

Punktacja

zad	pkt
1	0,5
2	1
3	1
4	1
5	3
6	3
7	6
8	4
razem	20