

# Raport

## Przetwarzanie i analiza danych przestrzennych

### Oracle spatial

Imiona i nazwiska: **Bartłomiej Chwast, Jakub Domogała**

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)

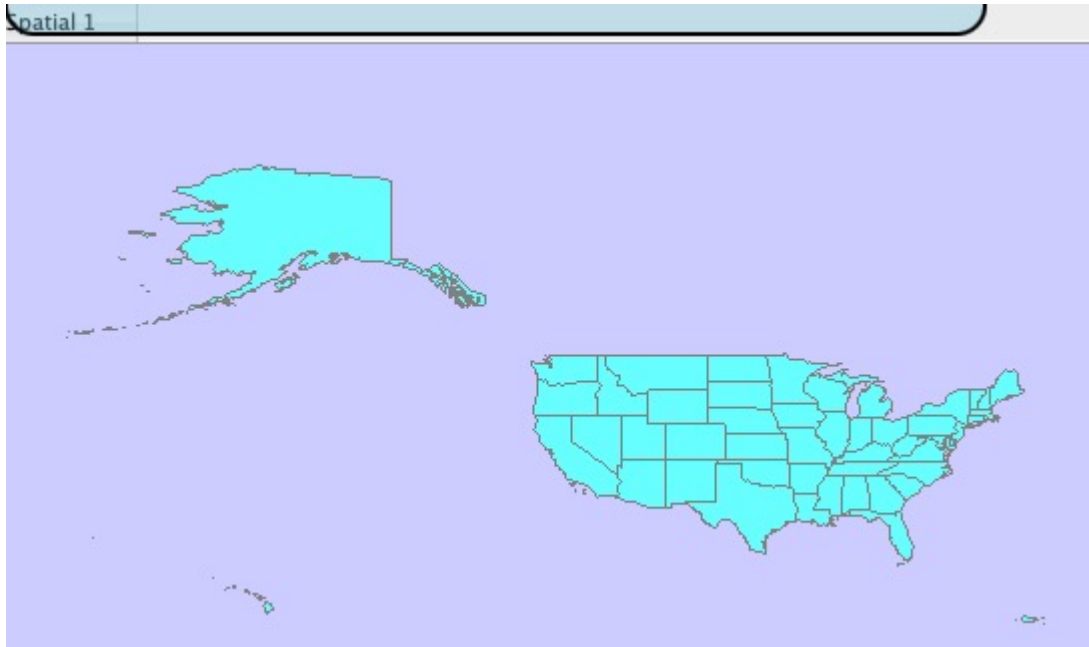
Zwróć uwagę na formatowanie kodu



# Zadanie 1

Zwizualizuj przykładowe dane

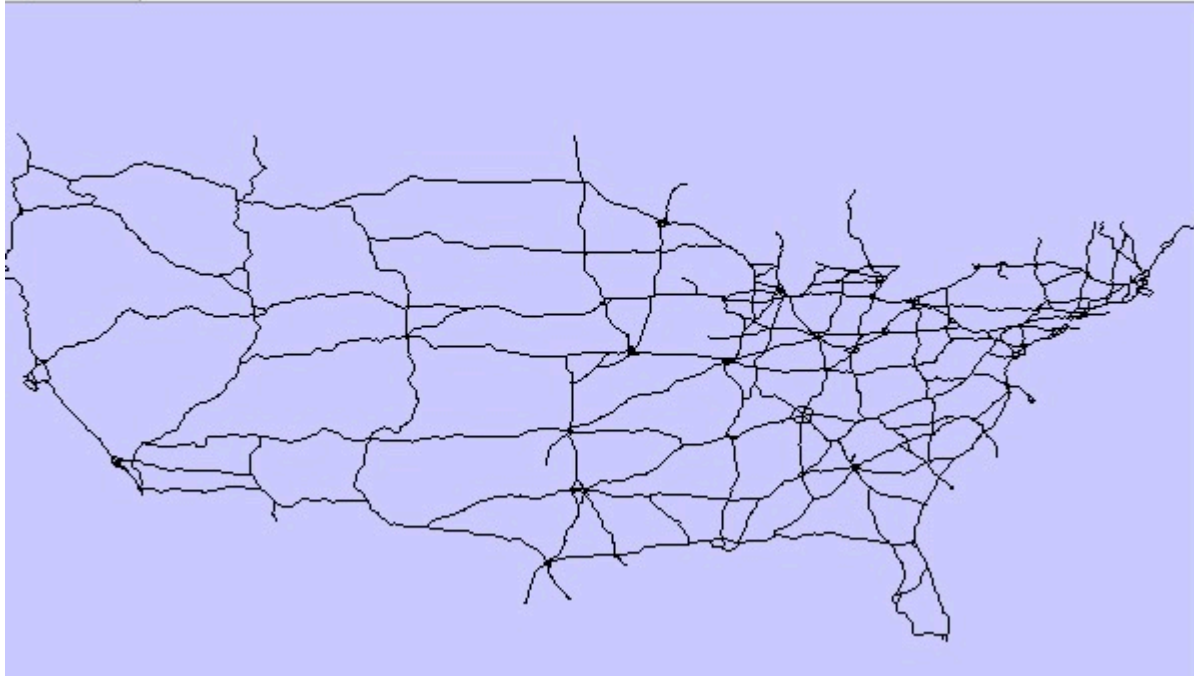
US\_STATES



```
SELECT * FROM us_states;
```

50 stanów USA

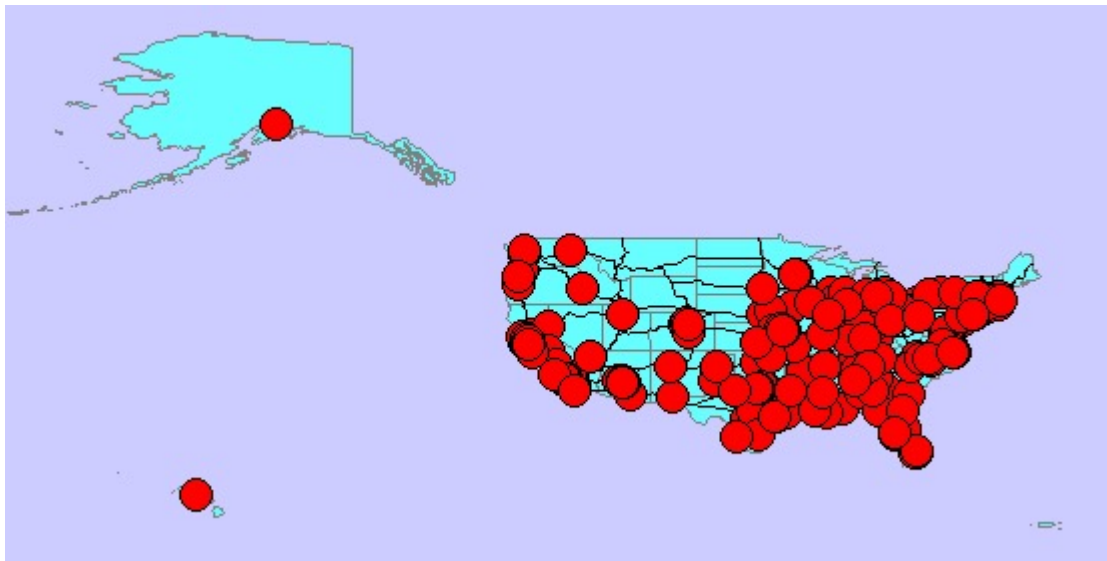
US\_INTERSTATES



```
SELECT * FROM us_interstates;
```

Autostrady międzystanowe są skoncentrowane w miejscach o większej gęstości zaludnienia, możemy to zauważyć na mapie, poprzez gęstszą siatkę w północno-wschodniej części kraju oraz na samym zachodnim wybrzeżu.

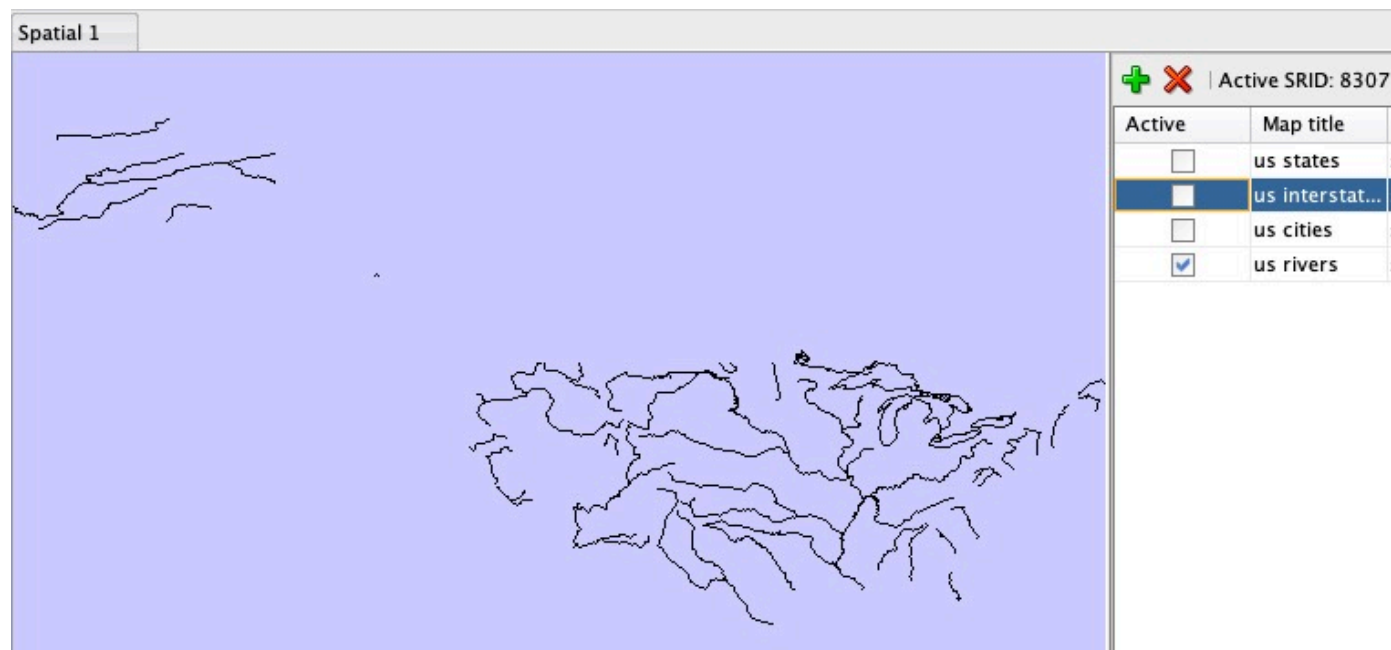
US\_CITIES



```
SELECT * FROM us_cities;
```

Miasta są skoncentrowane wzdłuż wybrzeży, wzdłuż rzek oraz w miejscach o większej gęstości zaludnienia.

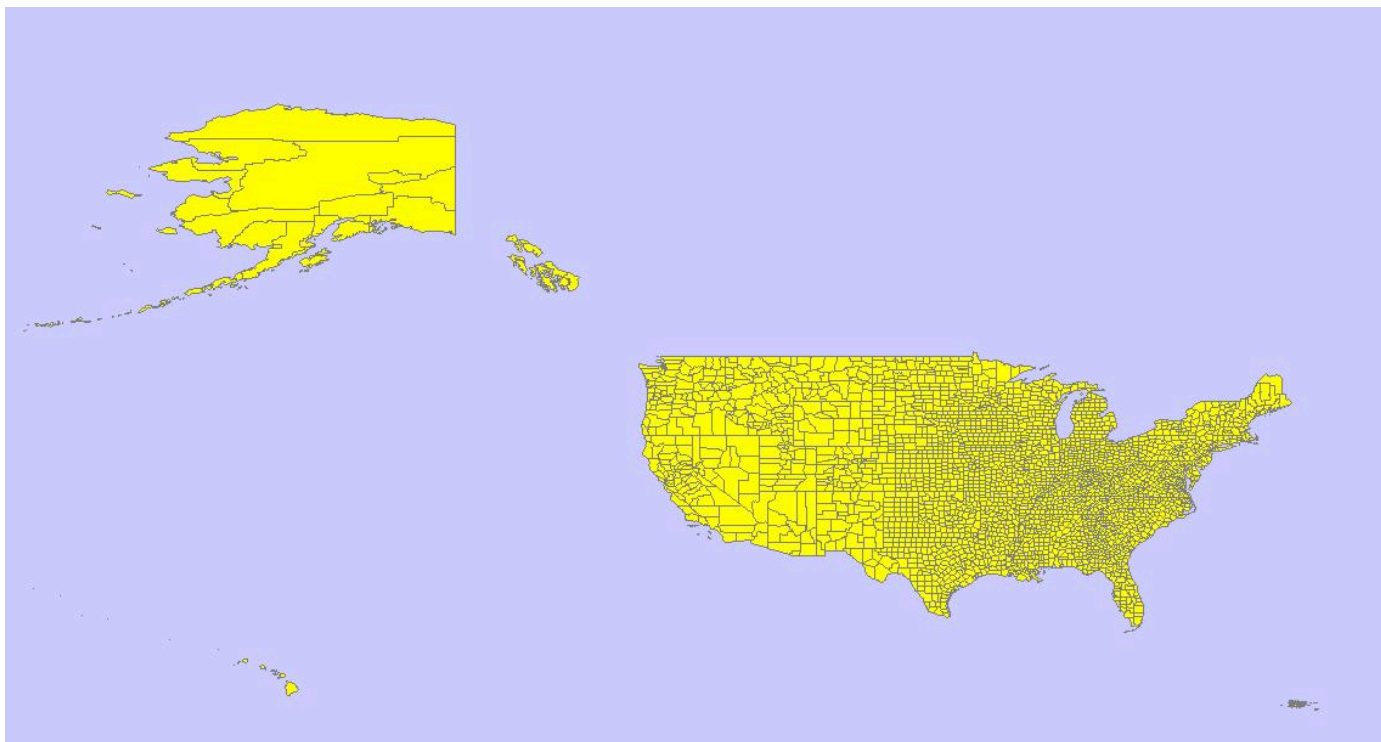
## US\_RIVERS



```
SELECT * FROM us_rivers;
```

Więcej rzek znajduje się w północno-wschodniej oraz w środkowej części kraju.

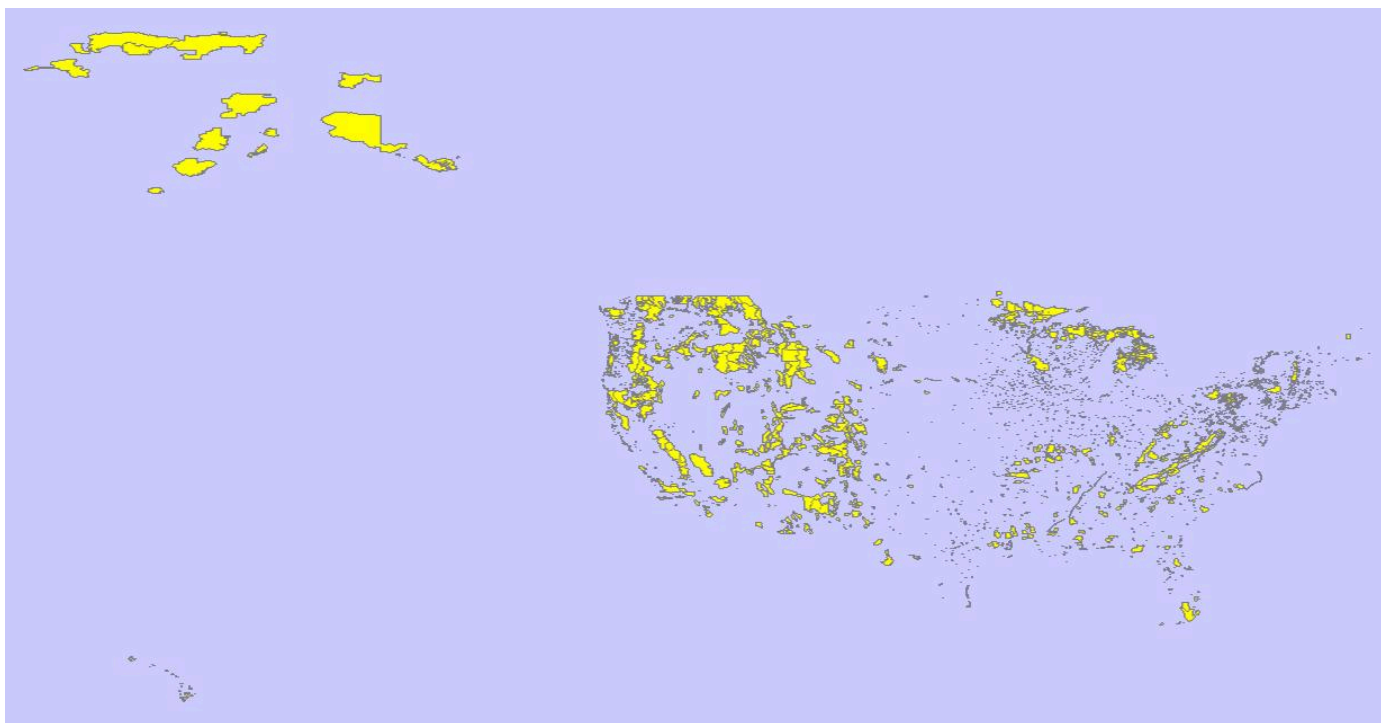
## US\_COUNTIES



```
SELECT * FROM us_counties;
```

Hrabstwa są liczniejsze w miejscach o większej gęstości zaludnienia, widać to na przykład w północno-wschodniej części kraju.

US\_PARKS



```
SELECT * FROM us_parks;
```

Parki narodowe są skoncentrowane w miejscach o mniejszej gęstości zaludnienia, widać to na przykład w północno-zachodniej części kraju.

## Zadanie 2

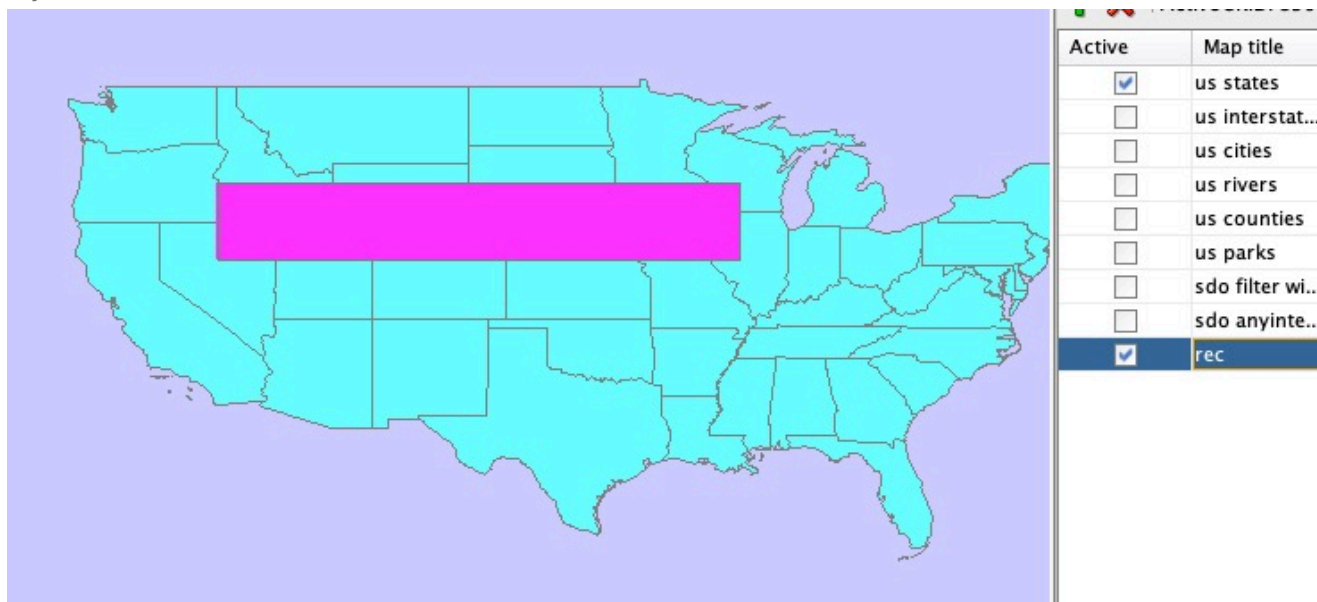
Znajdź wszystkie stany (us\_states) których obszary mają część wspólną ze wskazaną geometrią (prostokątem)

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

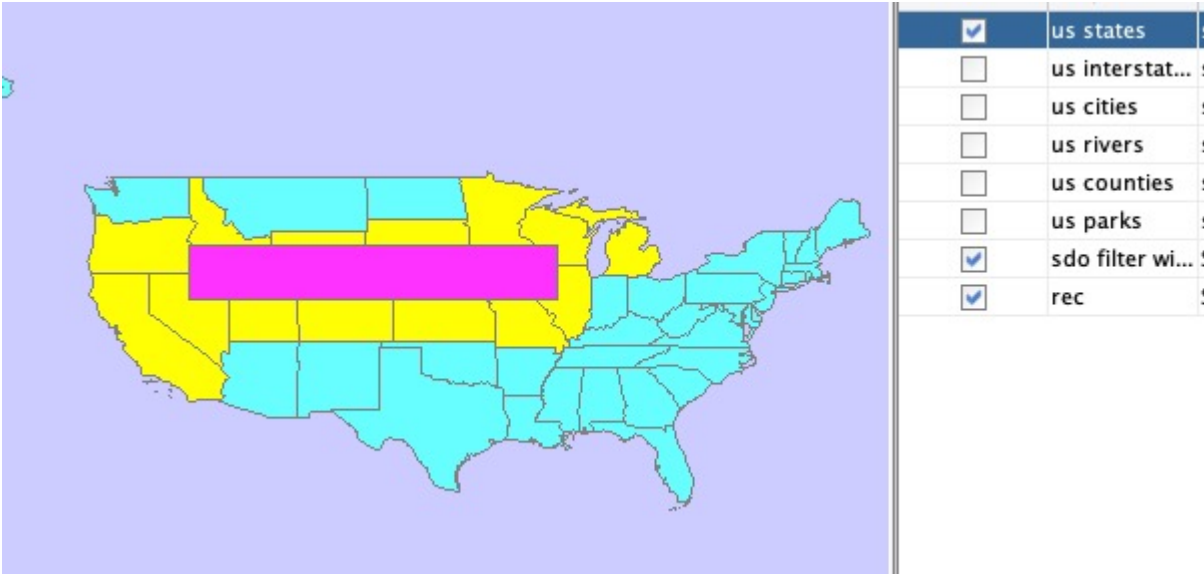


Użyj funkcji SDO\_FILTER

```
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



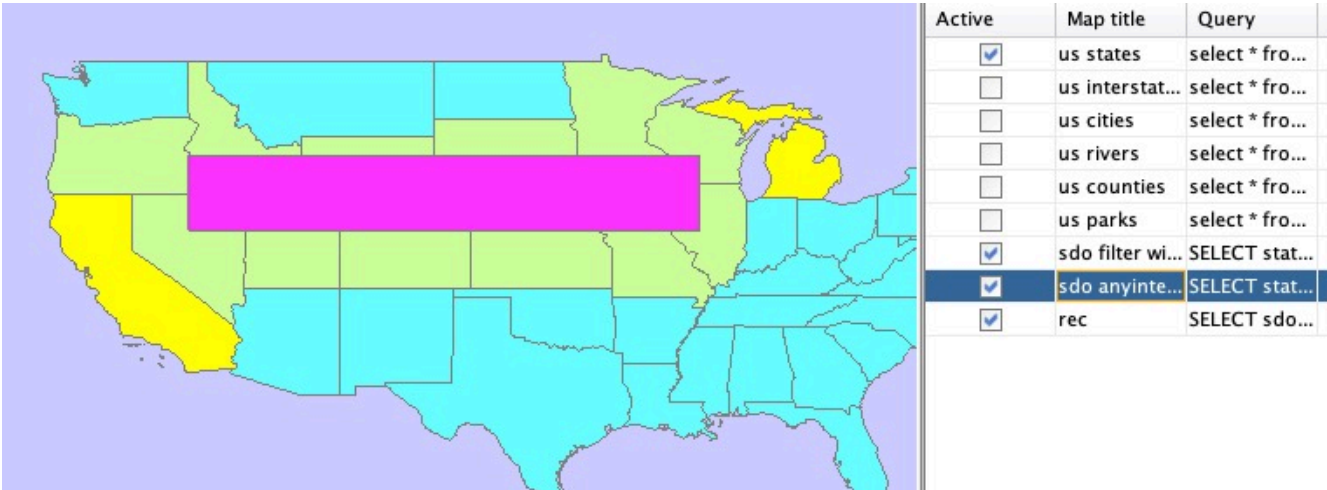
Użyj funkcji SDO\_ANYINTERACT

```
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





na żółto widzimy wystające części wyniku funkcji `filter`,  
zielonawy kolor przedstawia wyniki `anyinteract`

funkcja `sdo_anyinteract` daje lepsze wyniki,  
funkcja `filter` znajduje stany które w ogóle nie dotyczą naszego prostokąta.

Dzieje się tak ponieważ funkcja `filter` robi filtracje na bazie bounding box'u,  
a nie dokładnej geometrii obiektów.

Jest to obliczeniowo o wiele lżejszy proces, ale nie da nam idealnych wyników.

Potencjalnie najbardziej optymalnym podejściem wydaje się najpierw użycie funkcji `filter`

(bo jest szybka) na całym zbiorze danych, a następnie funkcji `anyinteract` na wyniku funkcji `filter`.

To pozwoliłoby nam wykonać najmniej obliczeń przy zachowaniu maksymalnej dokładności wyników.

---

## 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';
```

	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]
6	Casper Mtn County Park	[MDSYS.SDO_GEOMETRY]
7	City Park	[MDSYS.SDO_GEOMETRY]
8	Prosinski Park	[MDSYS.SDO_GEOMETRY]
9	Optimist Park	[MDSYS.SDO_GEOMETRY]
10	Undine Park	[MDSYS.SDO_GEOMETRY]
11	Washington Park	[MDSYS.SDO_GEOMETRY]
12	Municipal Park	[MDSYS.SDO_GEOMETRY]
13	Ayres Natural Bridge Park	[MDSYS.SDO_GEOMETRY]
14	Lewis Park	[MDSYS.SDO_GEOMETRY]
15	Fort Laramie NHS	[MDSYS.SDO_GEOMETRY]
16	Robinson Park	[MDSYS.SDO_GEOMETRY]
17	Pioneer Park	[MDSYS.SDO_GEOMETRY]
18	Jirdon Park	[MDSYS.SDO_GEOMETRY]
19	Jirdon Park	[MDSYS.SDO_GEOMETRY]
20	Devils Tower NMON	[MDSYS.SDO_GEOMETRY]
21	City Park	[MDSYS.SDO_GEOMETRY]
22	Washington Memorial Park	[MDSYS.SDO_GEOMETRY]
23	Fossil Butte NMON	[MDSYS.SDO_GEOMETRY]
24	Mark Iii M.H. Park	[MDSYS.SDO_GEOMETRY]
25	Teton NF	[MDSYS.SDO_GEOMETRY]
26	Grand Teton NP	[MDSYS.SDO_GEOMETRY]
27	John D. Rockefeller, Jr. Memor	[MDSYS.SDO_GEOMETRY]
28	Island Park	[MDSYS.SDO_GEOMETRY]
29	Diversion Dam Park	[MDSYS.SDO_GEOMETRY]
30	Pete Miller Park	[MDSYS.SDO_GEOMETRY]
31	Popo Agie Park	[MDSYS.SDO_GEOMETRY]
32	North Side Park	[MDSYS.SDO_GEOMETRY]

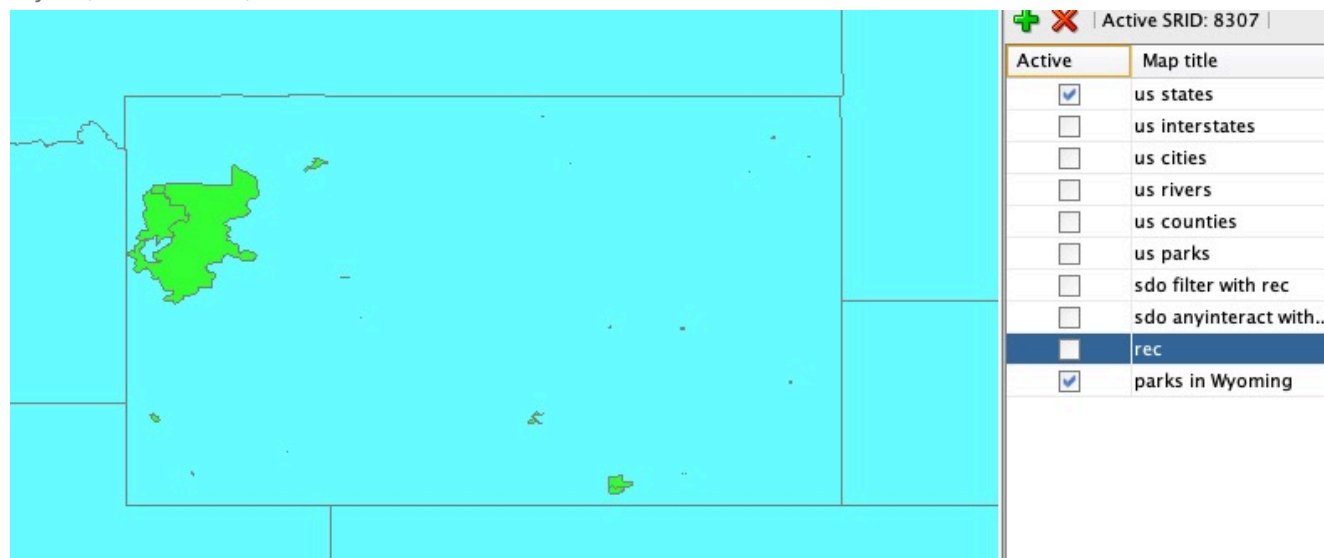
W przypadku wykorzystywania narzędzia SQL Developer, w celu wizualizacji na mapie użyj podzapytania

```

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'
)

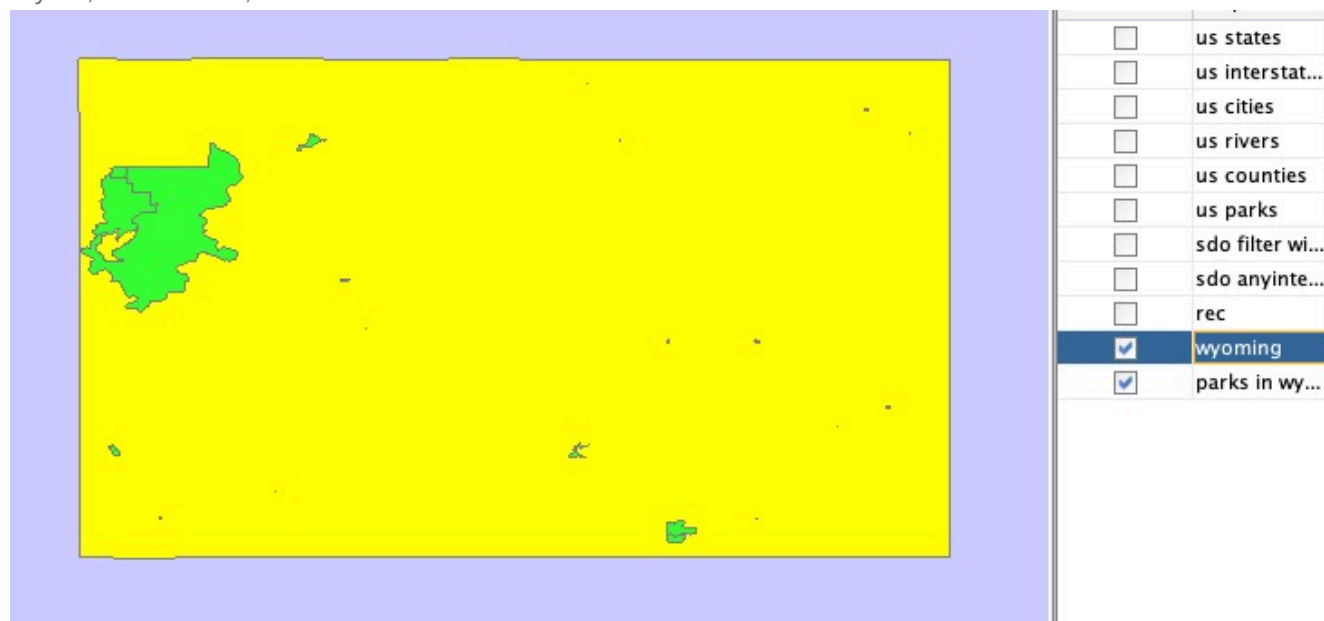
```

Wyniki, zrzut ekranu, komentarz



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

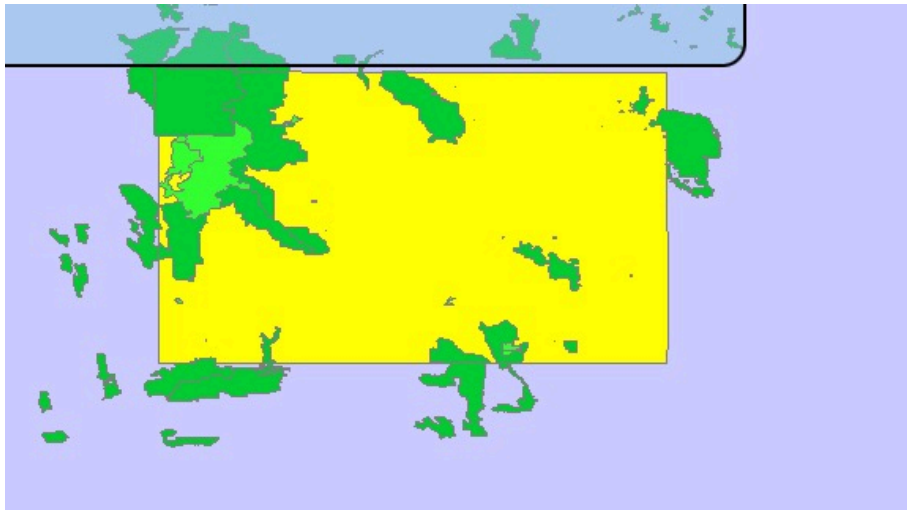
Wyniki, zrzut ekranu, komentarz



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



<input type="checkbox"/>	us states	select * fro...
<input type="checkbox"/>	us interstat...	select * fro...
<input type="checkbox"/>	us cities	select * fro...
<input type="checkbox"/>	us rivers	select * fro...
<input type="checkbox"/>	us counties	select * fro...
<input type="checkbox"/>	us parks	select * fro...
<input type="checkbox"/>	sdo filter wi...	SELECT stat...
<input type="checkbox"/>	sdo anyinte...	SELECT stat...
<input type="checkbox"/>	rec	SELECT sdo...
<input checked="" type="checkbox"/>	wyoming	SELECT stat...
<input checked="" type="checkbox"/>	parks inter...	select pp.n...
<input checked="" type="checkbox"/>	parks in wy...	SELECT pp....

Tak jak można się spodziewać funkcja anyinteract daje nam w wyniku nie tylko parki zawieające się w środku obszaru, ale również te które jedynie częściowo o ten obszar zahaczają.

```
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'
);
```

## Zadanie 4

Znajdź wszystkie jednostki administracyjne (us\_counties) wewnątrz stanu New Hampshire

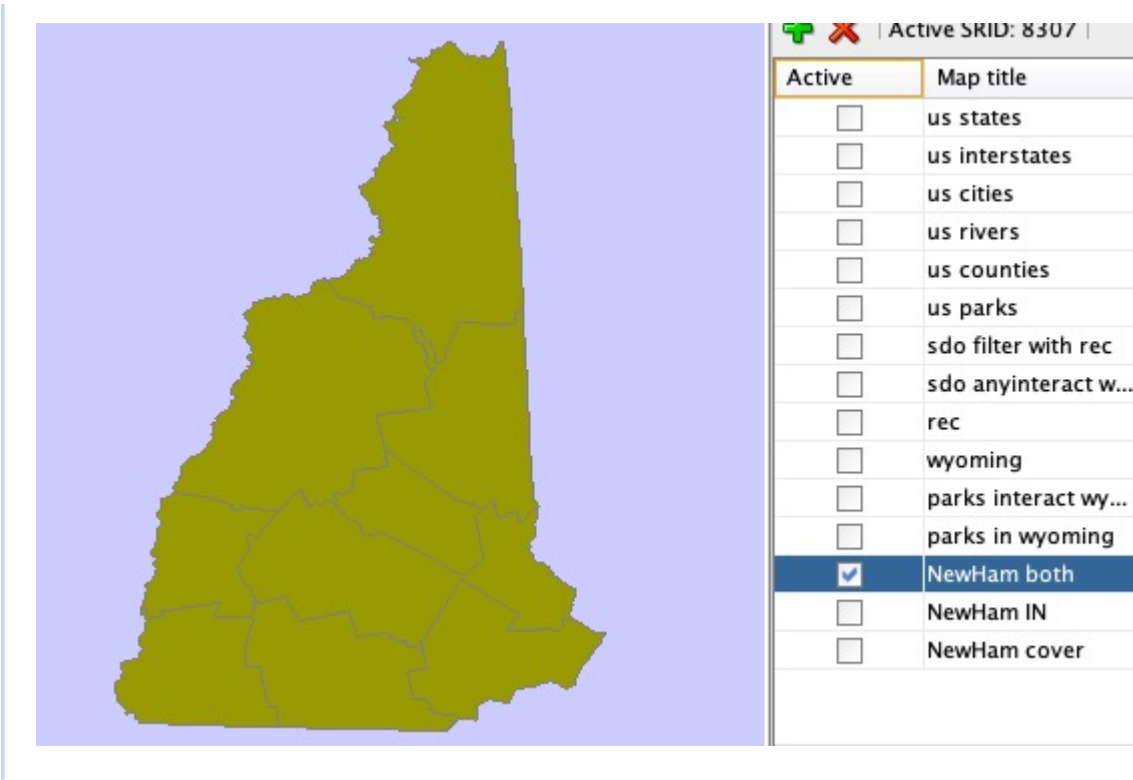
```
SELECT c.county, c.state_abrv, c.geom
FROM us_counties c, us_states s
WHERE s.state = 'New Hampshire'
AND SDO_RELATE ( c.geom,s.geom, 'mask=INSIDE+COVEREDBY') = 'TRUE';
```

```
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';
```

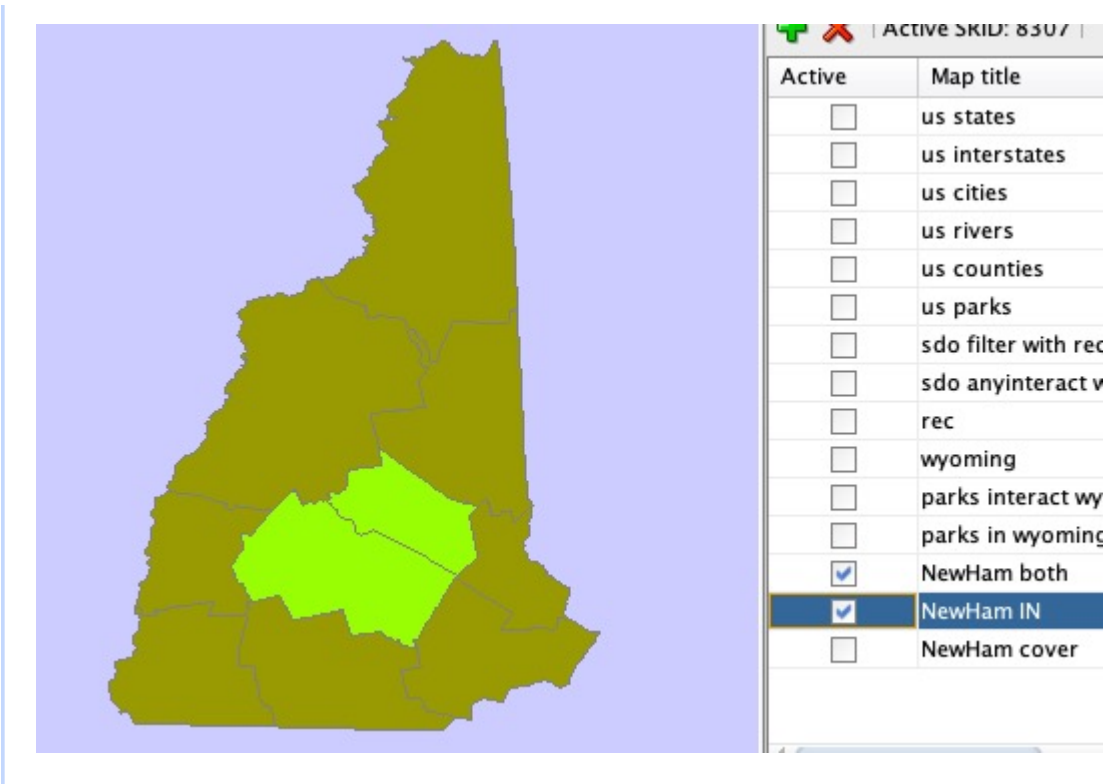
```
SELECT c.county, c.state_abrv, c.geom
FROM us_counties c, us_states s
WHERE s.state = 'New Hampshire'
AND SDO_RELATE ( c.geom,s.geom, 'mask=COVEREDBY') = 'TRUE';
```

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

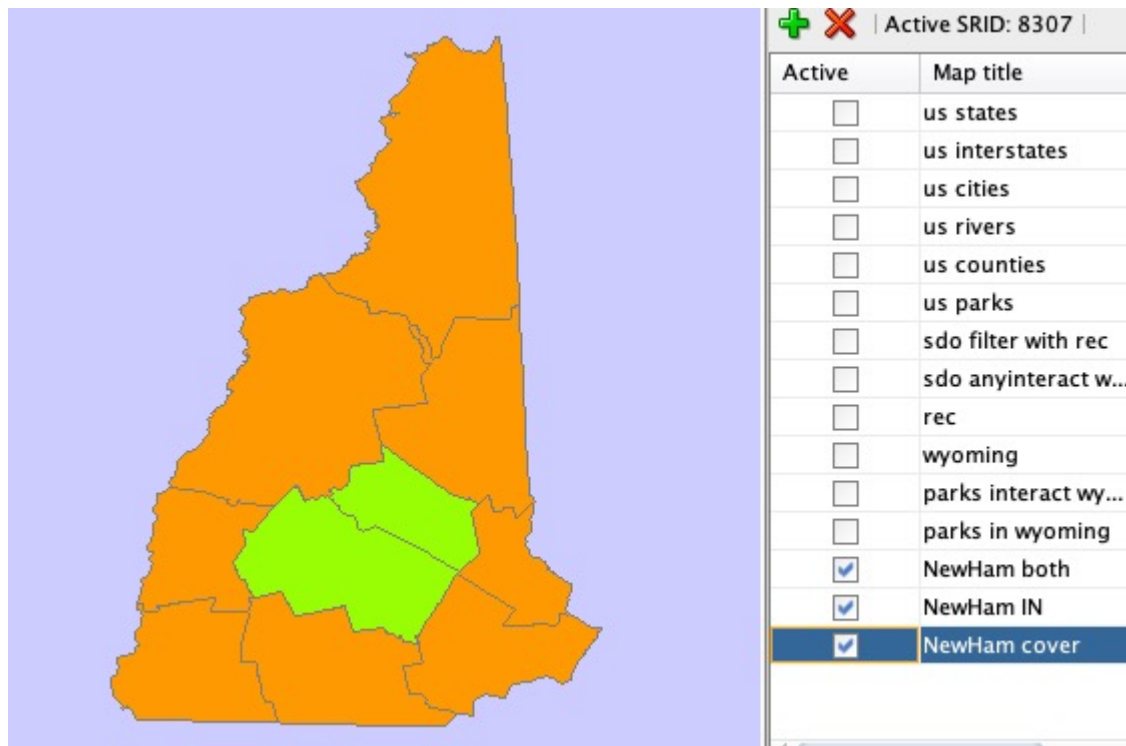
Obie maski



Maska Inside



Maska CoveredBy



Maski dają nam jako wyniki zbiory rozłączne które dają nam jako sumę ten sam obszar, co użycie ich obu w osobnych zapytaniach.

Co jednak dziwne to funkcja coveredby powinna zwracać pełny zbiór zwracany przez INSIDE + dodatkowe elementy dotyczące granic obszaru, jednak tak się nie dzieje. Z tego można wnioskować że funkcja covered by zwraca tylko wyniki dotyczące granic z wyłączeniem tych które są wewnątrz ale granic nie dotykają.

## Zmodyfikowane zapytania

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

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

```
SELECT pp.county, pp.state_abrv, pp.geom from us_counties pp
where id IN
(
    select c.id
    FROM us_counties c, us_states s
```

```
WHERE s.state = 'New Hampshire'
AND SDO_RELATE ( c.geom,s.geom, 'mask=COVEREDBY') = 'TRUE'
);
```

## 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'
)
```

```
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') = 'TRUE'
);
```

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



Dodatkowo:

a) Znajdz wszystkie jednostki administracyjne przez które przechodzi droga I4

```

SELECT c.county, c.state_abrv, c.geom
FROM us_counties c
WHERE ROWID IN
(
  SELECT c.rowid
  FROM us_counties c, us_interstates i
  WHERE SDO_ANYINTERACT(c.geom, i.geom) = 'TRUE'
  AND i.interstate = 'I4'
);

```



	COUNTY	STATE_ABRV	GEOM
1	Osceola	FL	[MDSYS.SDO_GEOMETRY]
2	Polk	FL	[MDSYS.SDO_GEOMETRY]
3	Hillsborough	FL	[MDSYS.SDO_GEOMETRY]
4	Orange	FL	[MDSYS.SDO_GEOMETRY]
5	Seminole	FL	[MDSYS.SDO_GEOMETRY]
6	Volusia	FL	[MDSYS.SDO_GEOMETRY]



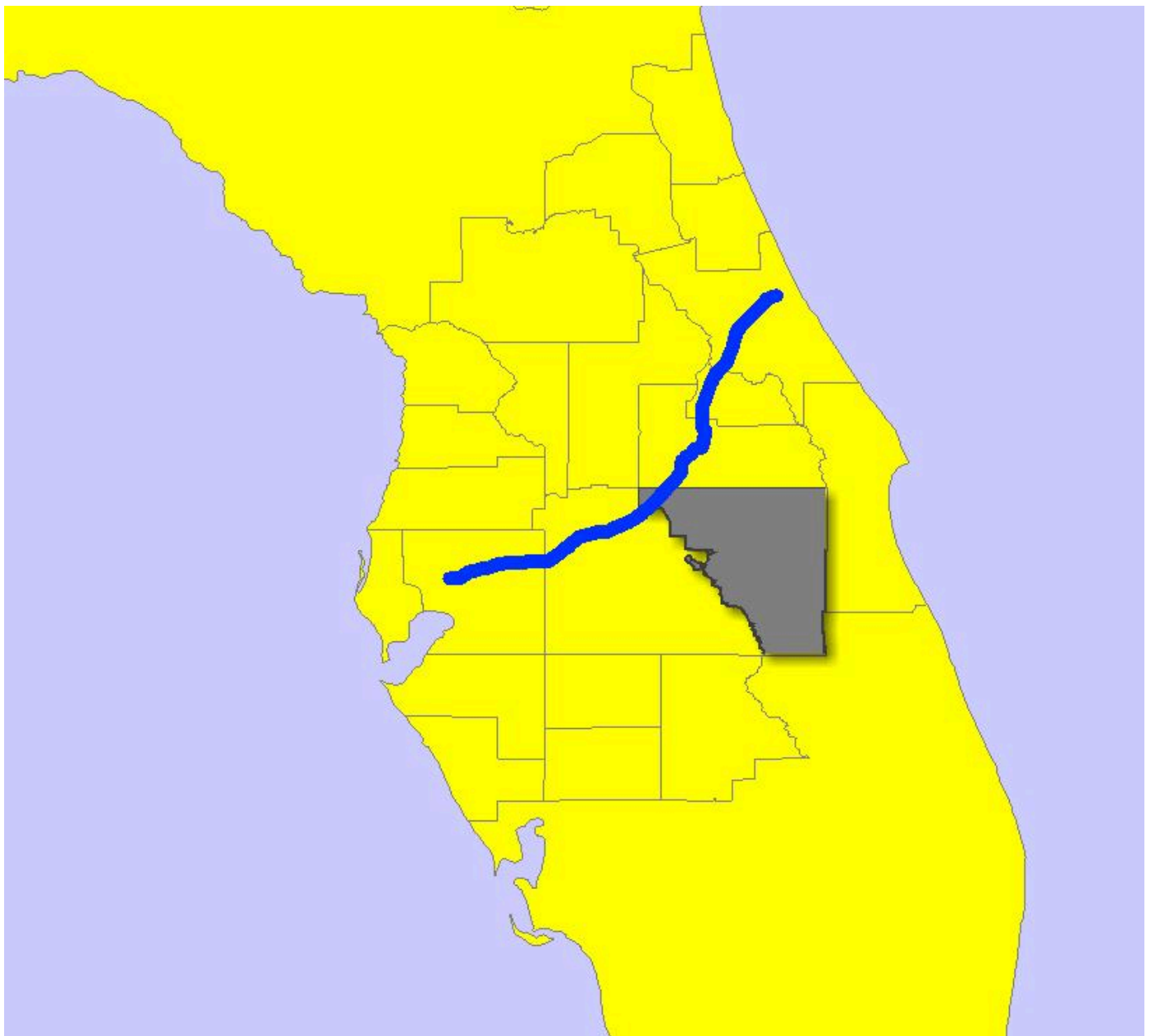
b) Znajdz wszystkie jednostki administracyjne w pewnej odległości od I4

```

SELECT c.county, c.state_abrv, c.geom
FROM us_counties c
WHERE ROWID IN
(
    SELECT c.rowid
    FROM us_counties c, us_interstates i
    WHERE SDO_WITHIN_DISTANCE(c.geom, i.geom, 'distance=50 unit=mile') = 'TRUE'
    AND i.interstate = 'I4'
);

```

	COUNTY	STATE_ABRV	GEOM
1	Manatee	FL	[MDSYS.SDO_GEOMETRY]
2	Sarasota	FL	[MDSYS.SDO_GEOMETRY]
3	Hardee	FL	[MDSYS.SDO_GEOMETRY]
4	DeSoto	FL	[MDSYS.SDO_GEOMETRY]
5	Highlands	FL	[MDSYS.SDO_GEOMETRY]
6	Pinellas	FL	[MDSYS.SDO_GEOMETRY]
7	Osceola	FL	[MDSYS.SDO_GEOMETRY]
8	Polk	FL	[MDSYS.SDO_GEOMETRY]
9	Hillsborough	FL	[MDSYS.SDO_GEOMETRY]
10	Brevard	FL	[MDSYS.SDO_GEOMETRY]
11	Pasco	FL	[MDSYS.SDO_GEOMETRY]
12	Sumter	FL	[MDSYS.SDO_GEOMETRY]
13	Lake	FL	[MDSYS.SDO_GEOMETRY]
14	Orange	FL	[MDSYS.SDO_GEOMETRY]
15	Hernando	FL	[MDSYS.SDO_GEOMETRY]
16	Seminole	FL	[MDSYS.SDO_GEOMETRY]
17	Volusia	FL	[MDSYS.SDO_GEOMETRY]
18	Citrus	FL	[MDSYS.SDO_GEOMETRY]
19	Marion	FL	[MDSYS.SDO_GEOMETRY]
20	Flagler	FL	[MDSYS.SDO_GEOMETRY]
21	Putnam	FL	[MDSYS.SDO_GEOMETRY]
22	St. Johns	FL	[MDSYS.SDO_GEOMETRY]



c) Znajdz rzeki które przecina droga I4

```
SELECT r.name, r.geom
FROM us_rivers r
WHERE ROWID IN
(
  SELECT r.rowid
  FROM us_rivers r, us_interstates i
  WHERE SDO_ANYINTERACT(r.geom, i.geom) = 'TRUE'
  AND i.interstate = 'I4'
);
```

	NAME	GEOM	
1	St. Johns	[MDSYS.SDO_GEOMETRY]	



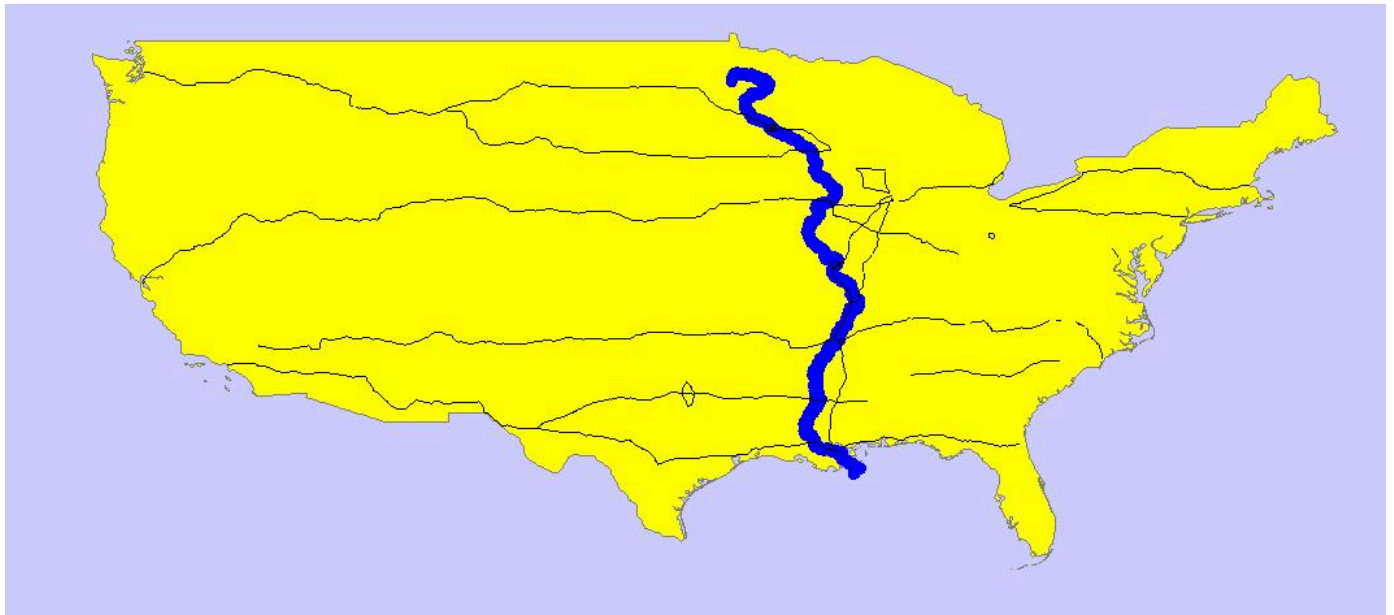
d) Znajdz wszystkie drogi które przecinają rzekę Mississippi

```

SELECT i.interstate, i.geom
FROM us_interstates i
WHERE ROWID IN
(
  SELECT i.rowid
  FROM us_rivers r, us_interstates i
  WHERE SDO_ANYINTERACT(r.geom, i.geom) = 'TRUE'
  AND r.name = 'Mississippi'
)

```

	INTERSTATE	GEOM
1	I40	[MDSYS.SDO_GEOMETRY]
2	I20	[MDSYS.SDO_GEOMETRY]
3	I10	[MDSYS.SDO_GEOMETRY]
4	I255	[MDSYS.SDO_GEOMETRY]
5	I270	[MDSYS.SDO_GEOMETRY]
6	I35E	[MDSYS.SDO_GEOMETRY]
7	I35W	[MDSYS.SDO_GEOMETRY]
8	I55	[MDSYS.SDO_GEOMETRY]
9	I55/I70	[MDSYS.SDO_GEOMETRY]
10	I57	[MDSYS.SDO_GEOMETRY]
11	I90	[MDSYS.SDO_GEOMETRY]
12	I94	[MDSYS.SDO_GEOMETRY]
13	I494	[MDSYS.SDO_GEOMETRY]
14	I74	[MDSYS.SDO_GEOMETRY]
15	I80	[MDSYS.SDO_GEOMETRY]



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

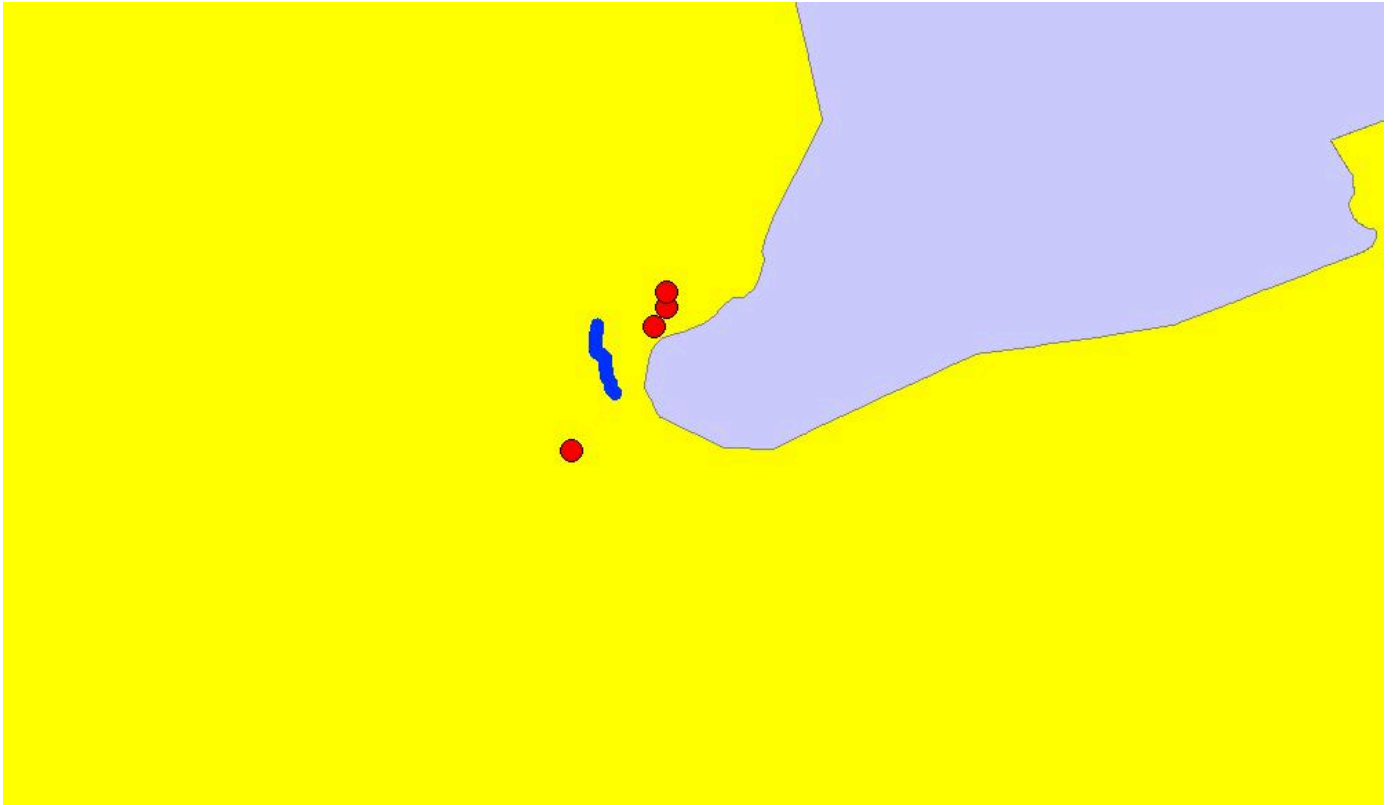
```

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'
    MINUS
    SELECT c.rowid
    FROM us_interstates i, us_cities c
    WHERE i.interstate = 'I275'
    AND sdo_within_distance (c.location, i.geom, 'distance=15 unit=mile') = 'TRUE'

```

)

	CITY	STATE_ABRV	LOCATION
1	Sterling Heights	MI	[MDSYS.SDO_GEOMETRY]
2	Detroit	MI	[MDSYS.SDO_GEOMETRY]
3	Toledo	OH	[MDSYS.SDO_GEOMETRY]
4	Warren	MI	[MDSYS.SDO_GEOMETRY]



f) Itp. (własne przykłady)

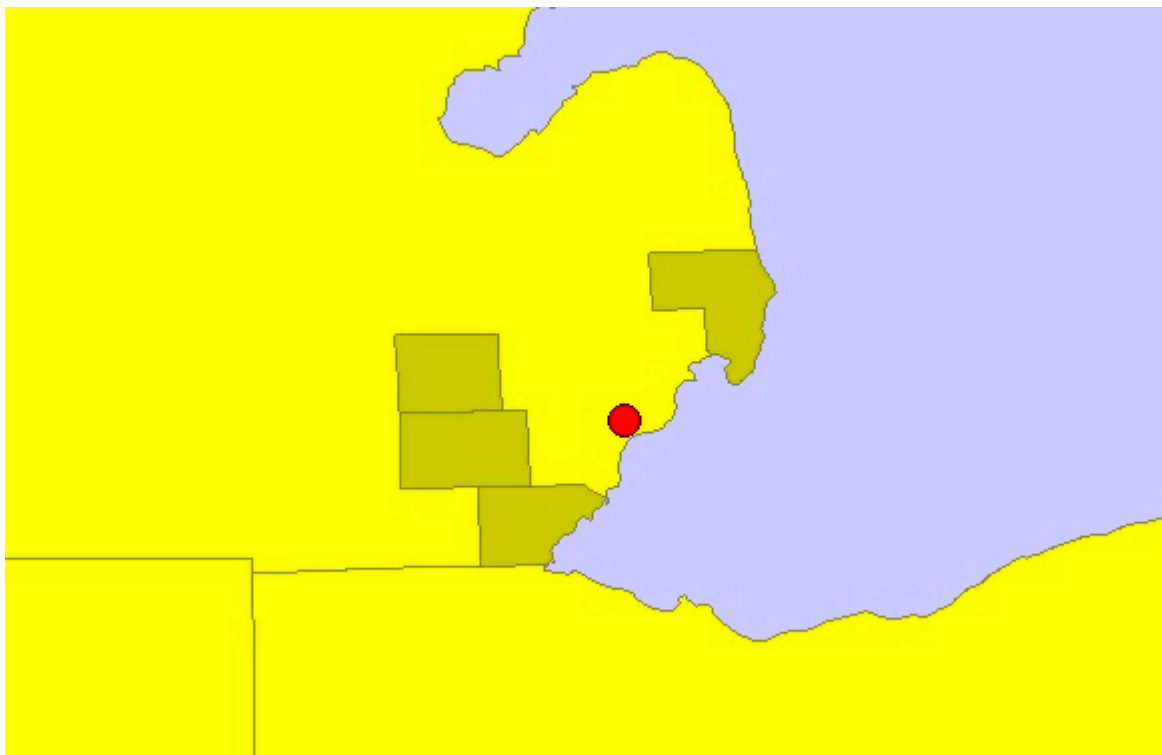
f) znajdź wszystkie jednostki administracyjne w odległości od 20 do 50 kilometrów od miasta Detroit

```
SELECT c.county, c.state_abrv, c.geom
FROM us_counties c
WHERE ROWID IN
(
  SELECT a.rowid
  FROM us_counties a, us_cities b
  WHERE SDO_WITHIN_DISTANCE(a.geom, b.location, 'distance=50 unit=KM') = 'TRUE'
  AND b.city = 'Detroit'
  MINUS

  SELECT a.rowid
  FROM us_counties a, us_cities b
  WHERE SDO_WITHIN_DISTANCE(a.geom, b.location, 'distance=20 unit=KM') = 'TRUE'
  AND b.city = 'Detroit'
```

)

	COUNTY	STATE_ABRV	GEOM
1	Monroe	MI	[MDSYS.SDO_GEOMETRY]
2	St. Clair	MI	[MDSYS.SDO_GEOMETRY]
3	Washtenaw	MI	[MDSYS.SDO_GEOMETRY]
4	Livingston	MI	[MDSYS.SDO_GEOMETRY]



## 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

```
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_nn(c.location, i.geom, 'sdo_num_res=5') = 'TRUE'
)

```

	CITY	STATE_ABRV	LOCATION
1	Tampa	FL	[MDSYS.SDO_GEOMETRY]
2	Jacksonville	FL	[MDSYS.SDO_GEOMETRY]
3	St Petersburg	FL	[MDSYS.SDO_GEOMETRY]
4	Orlando	FL	[MDSYS.SDO_GEOMETRY]
5	Fort Lauderdale	FL	[MDSYS.SDO_GEOMETRY]



Dodatkowo:



a) Znajdz kilka miast najbliższych rzece Mississippi

```
SELECT c.city, c.state_abrv, c.location
FROM us_cities c
WHERE ROWID IN
(
    SELECT c.rowid
    FROM us_rivers r, us_cities c
    WHERE r.name = 'Mississippi'
    AND sdo_nn(c.location, r.geom, 'sdo_num_res=5') = 'TRUE'
)
```

	CITY	STATE_ABRV	LOCATION
1	St Paul	MN	[MDSYS.SDO_GEOMETRY]
2	Memphis	TN	[MDSYS.SDO_GEOMETRY]
3	St Louis	MO	[MDSYS.SDO_GEOMETRY]
4	Minneapolis	MN	[MDSYS.SDO_GEOMETRY]
5	Baton Rouge	LA	[MDSYS.SDO_GEOMETRY]



b) Znajdz 3 miasta najbliższej Nowego Jorku

```

SELECT c.city, c.state_abrv, c.location
FROM us_cities c
WHERE ROWID IN
(
  SELECT c.rowid
  FROM us_cities c
  WHERE c.city != 'New York'
  AND sdo_nn(c.location,
    (SELECT location FROM us_cities WHERE city = 'New York'),
    'sdo_num_res=4') = 'TRUE'
)

```

	CITY	STATE_ABRV	LOCATION	
1	Elizabeth	NJ	[MDSYS.SDO_GEOMETRY]	
2	Newark	NJ	[MDSYS.SDO_GEOMETRY]	
3	Jersey City	NJ	[MDSYS.SDO_GEOMETRY]	



c) Znajdz kilka jednostek administracyjnych (us\_counties) z których jest najbliżej do Nowego Jorku

```

SELECT c.county, c.state_abrv, c.geom
FROM us_counties c
WHERE ROWID IN
(

```

```

SELECT c.rowid
FROM us_counties c, us_cities ci
WHERE ci.city = 'New York'
AND sdo_nn(c.geom, ci.location, 'sdo_num_res=5') = 'TRUE'
)

```

	COUNTY	STATE_ABRV	GEOM
1	Kings	NY	[MDSYS.SDO_GEOMETRY]
2	Hudson	NJ	[MDSYS.SDO_GEOMETRY]
3	New York	NY	[MDSYS.SDO_GEOMETRY]
4	Queens	NY	[MDSYS.SDO_GEOMETRY]
5	Richmond	NY	[MDSYS.SDO_GEOMETRY]



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

```

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 = 'I170'
    AND sdo_nn(c.location, i.geom, 'sdo_num_res=5') = 'TRUE'
)

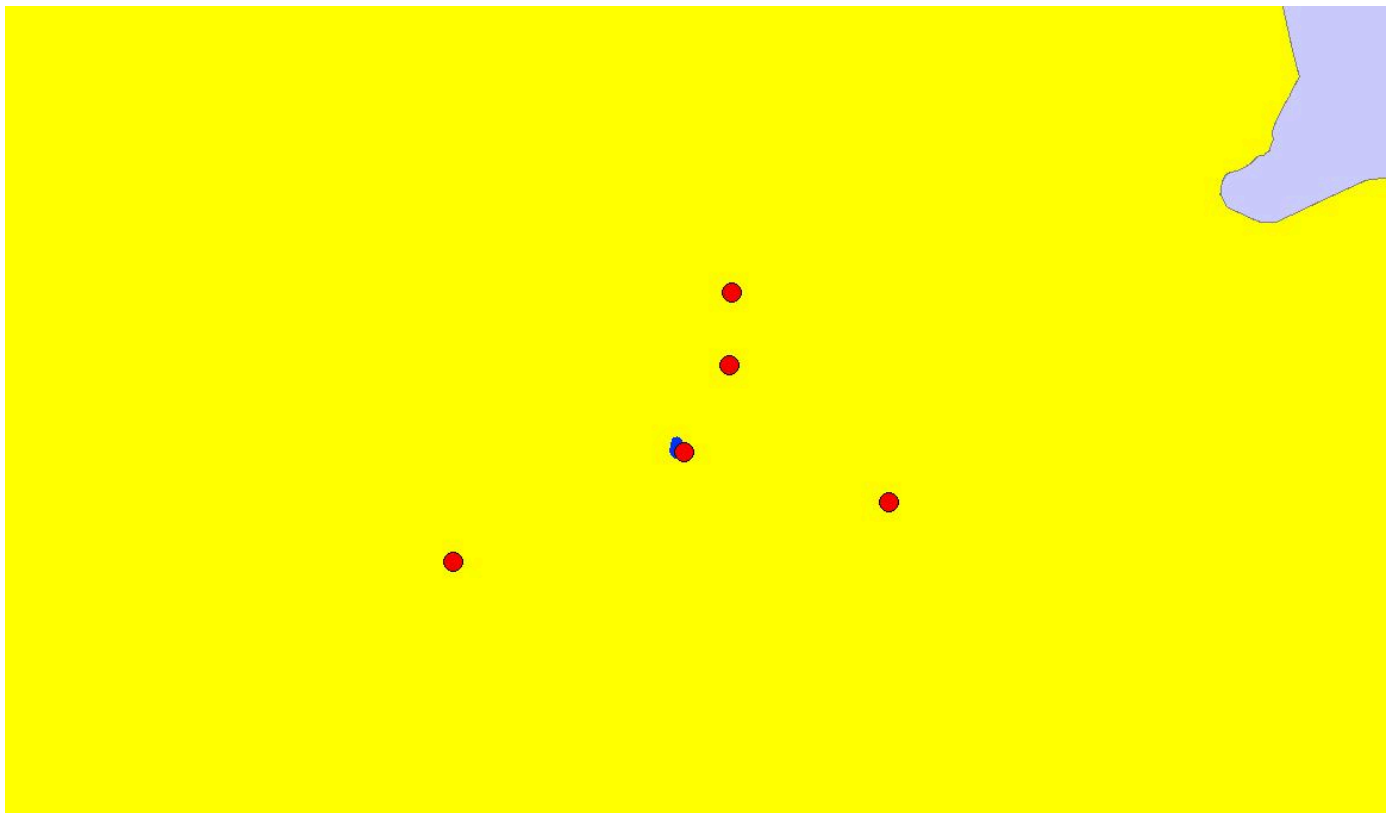
```

```

SELECT c.city, c.state_abrv, c.location,
       sdo_geom.sdo_distance(c.location, i.geom, 0.005, 'unit=KM') distance
FROM us_cities c, us_interstates i
WHERE i.interstate = 'I170'
AND sdo_nn(c.location, i.geom, 'sdo_num_res=5') = 'TRUE'
ORDER BY distance;

```

	CITY	STATE_ABRV	LOCATION	DISTANCE
1	St Louis	MO	[MDSYS.SDO_GEOMETRY]	8.63086834124045
2	Springfield	IL	[MDSYS.SDO_GEOMETRY]	126.815899024404
3	Peoria	IL	[MDSYS.SDO_GEOMETRY]	227.686805598134
4	Evansville	IN	[MDSYS.SDO_GEOMETRY]	254.637198689794
5	Springfield	MO	[MDSYS.SDO_GEOMETRY]	303.375234373403



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

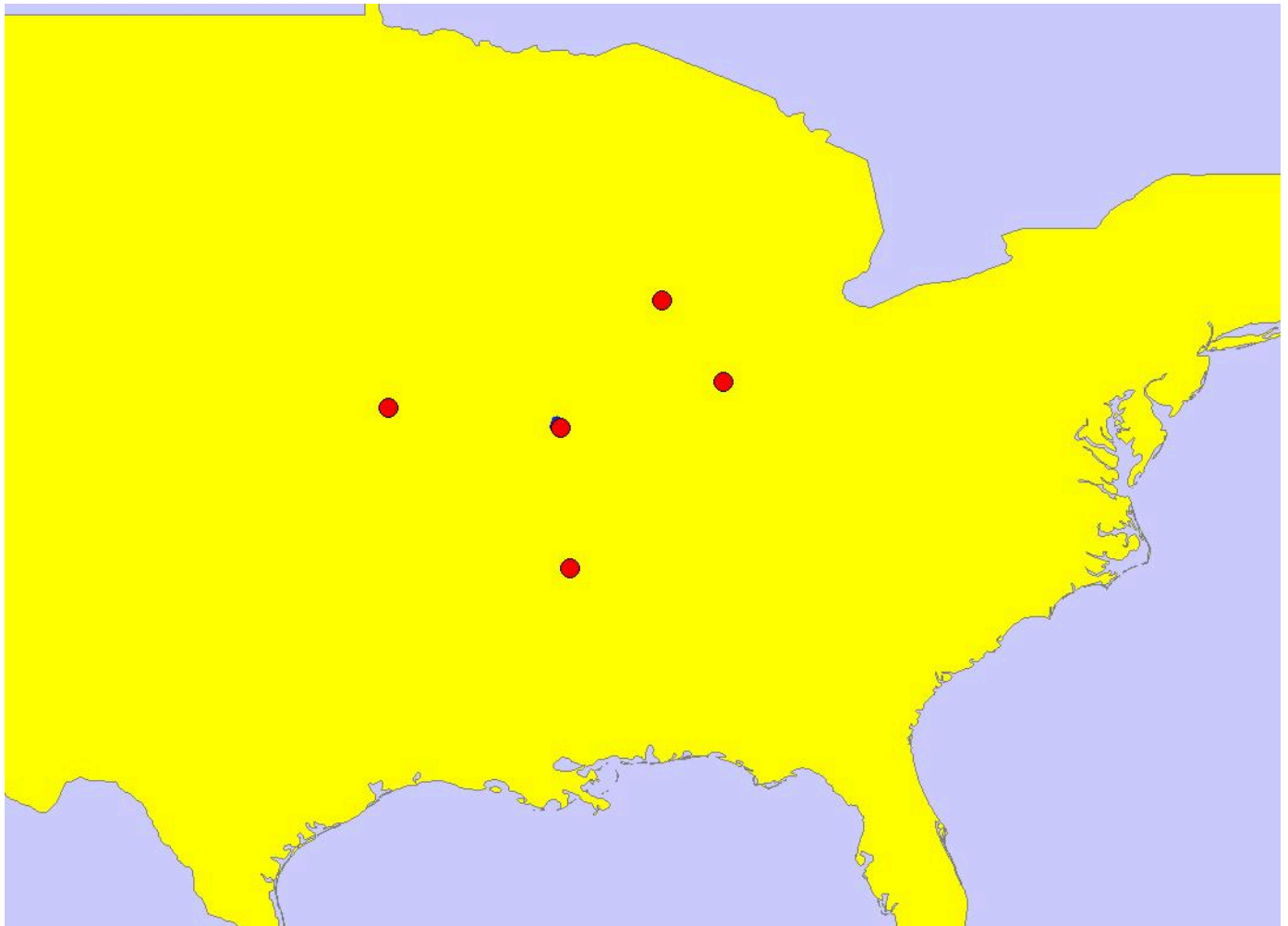
```

SELECT c.city, c.state_abrv, c.location
FROM us_cities c
WHERE ROWID IN
(
  SELECT r FROM
  (
    SELECT c.rowid r, c.city, sdo_geom.sdo_distance(c.location, i.geom, 0.005, 'unit=KM') distance
    FROM us_cities c, us_interstates i
    WHERE i.interstate = 'I170' AND c.pop90 > 300000
    ORDER BY distance
  )
)
WHERE ROWNUM <= 5

```

)

	CITY	STATE_ABRV	LOCATION
1	Kansas City	MO	[MDSYS.SDO_GEOMETRY]
2	Memphis	TN	[MDSYS.SDO_GEOMETRY]
3	Indianapolis	IN	[MDSYS.SDO_GEOMETRY]
4	Chicago	IL	[MDSYS.SDO_GEOMETRY]
5	St Louis	MO	[MDSYS.SDO_GEOMETRY]



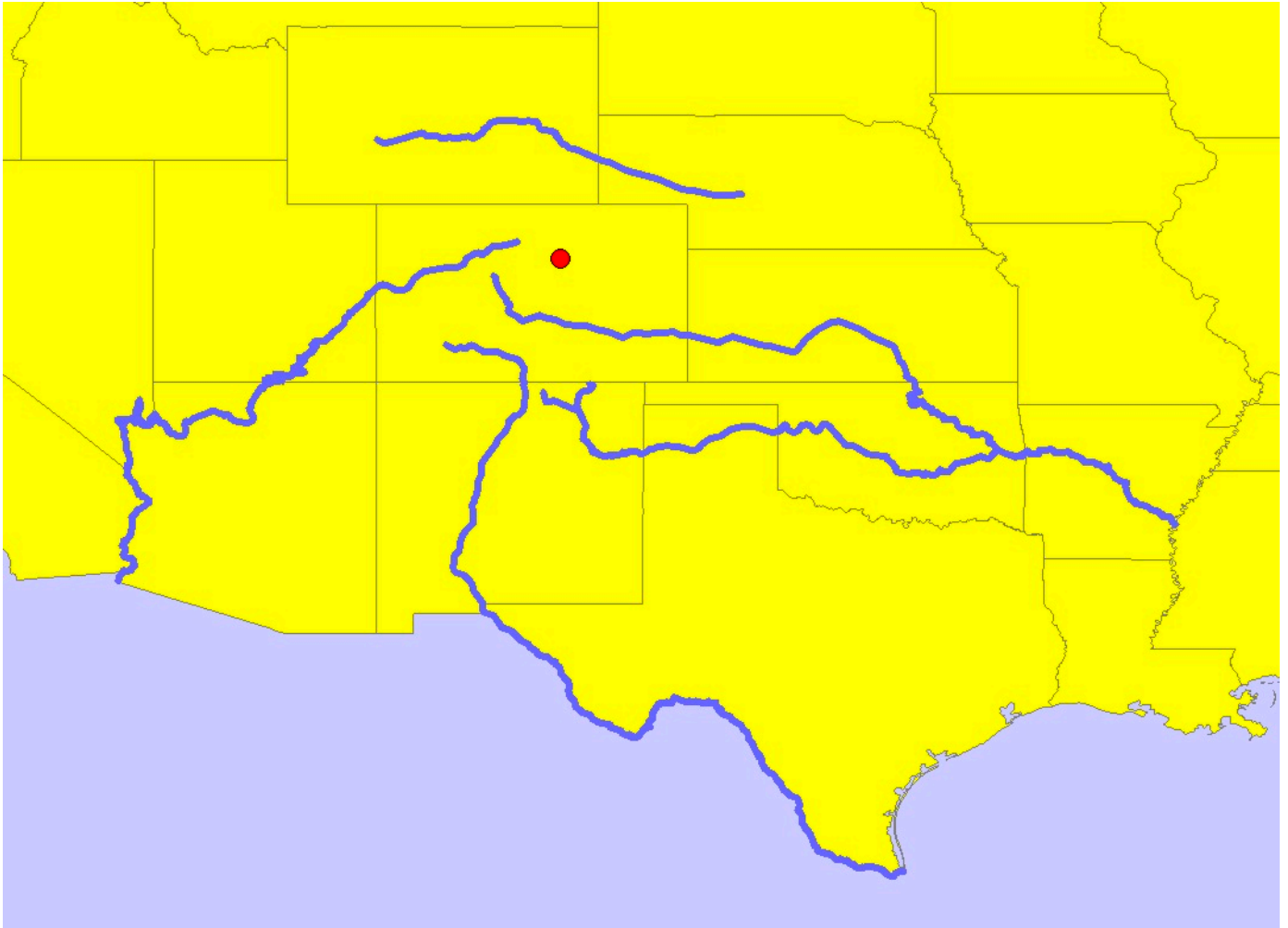
f) ltp. (własne przykłady)

f) Znajdź 5 najbliższych rzek od Denver

```
SELECT r.name, r.geom
FROM us_rivers r
WHERE ROWID IN
(
  SELECT r.rowid
  FROM us_rivers r, us_cities c
  WHERE c.city = 'Denver'
  AND sdo_nn(r.geom, c.location, 'sdo_num_res=5') = 'TRUE'
```

)

	NAME	GEOM
1	Colorado	[MDSYS.SDO_GEOMETRY]
2	North Platte	[MDSYS.SDO_GEOMETRY]
3	Arkansas	[MDSYS.SDO_GEOMETRY]
4	Rio Grande	[MDSYS.SDO_GEOMETRY]
5	Canadian	[MDSYS.SDO_GEOMETRY]



## Zadanie 7

Oblicz długość drogi I4

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

	LENGTH
1	212.260756199927

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

b) Która droga jest najdłuższa/najkrótsza

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

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

	INTERSTATE	LENGTH
1	I90	4290.6462617249

	INTERSTATE	LENGTH
1	I564	0.462140186764249

c) Która rzeka jest najdłuższa/najkrótsza

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

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

	NAME	LENGTH
1	St. Clair	6950.91937515048

	NAME	LENGTH
1	Richelieu	1.16169766454518

d) Które stany mają najdłuższą granicę

```
SELECT state, SDO_GEOM.SDO_LENGTH (geom, 0.5, 'unit=kilometer') length
FROM us_states
ORDER BY length DESC
FETCH FIRST 5 ROW ONLY;
```

	STATE	LENGTH
1	Alaska	26138.3745019651
2	Texas	6779.84795094551
3	California	4145.76647746918
4	Michigan	4140.12257047995
5	Florida	3725.07858238253

e) Itp. (własne przykłady)

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

-- ...

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';
```

	DISTANCE
1	222184.610363969

Dodatkowo:

a) Oblicz odległość między miastem Tampa a drogą I4

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

	DISTANCE
1	3103.91172130556



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

```
SELECT SDO_GEOM.SDO_DISTANCE(s1.geom, s2.geom, 0.5) distance
FROM us_states s1, us_states s2
WHERE s1.state = 'New York' AND s2.state = 'Florida';
```

	DISTANCE
1	1256583.87785727

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

```
SELECT SDO_GEOM.SDO_DISTANCE(s.geom, c.location, 0.5) distance
FROM us_states s, us_cities c
WHERE c.city = 'New York' AND s.state = 'Florida';
```

	DISTANCE
1	1296590.76150732

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

```
SELECT p.name, SDO_GEOM.SDO_DISTANCE(p.geom, c.location, 0.5) distance
FROM us_parks p, us_cities c
WHERE c.city = 'New York'
ORDER BY distance
FETCH FIRST 3 ROWS ONLY;
```

	NAME	DISTANCE
1	Institute Park	1539.89392335604
2	Prospect Park	1718.06926034585
3	Thompkins Park	2135.55672310316

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

-- ...

#### Punktacja

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