

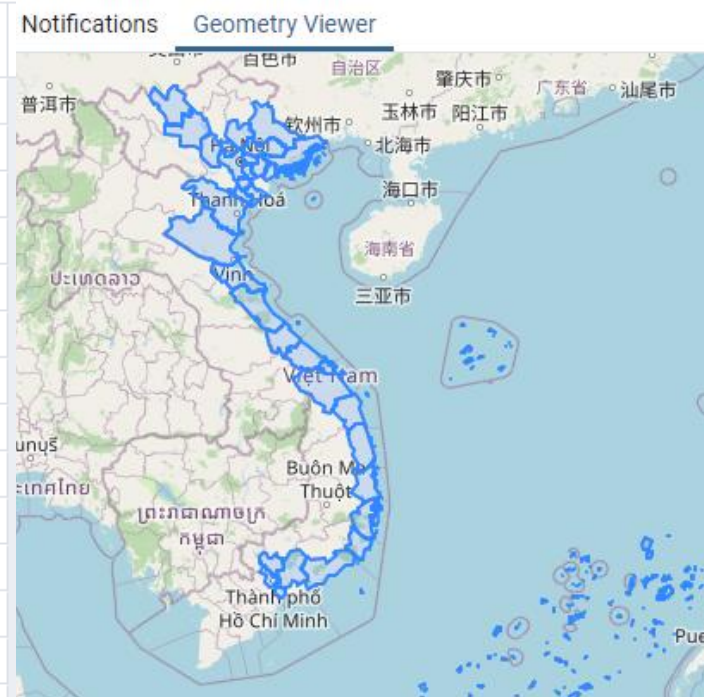


# Truy vấn dữ liệu không gian trong cơ sở dữ liệu địa lý

4



	ten	geom
	character varying (254)	geometry
1	Tỉnh Đồng Nai	0106000020E6100...
2	Tỉnh Ninh Bình	0106000020E6100...
3	Tỉnh Bình Dương	0106000020E6100...
4	Thành phố Hà Nội	0106000020E6100...
5	Tỉnh Nghệ An	0106000020E6100...
6	Tỉnh Quảng Trị	0106000020E6100...
7	Tỉnh Bình Thuận	0106000020E6100...
8	Tỉnh Yên Bái	0106000020E6100...
9	Tỉnh Thái Nguyên	0106000020E6100...
10	Tỉnh Thanh Hóa	0106000020E6100...
11	Tỉnh Hà Tĩnh	0106000020E6100...
12	Thành phố Đà Nẵng	0106000020E6100...
13	Tỉnh Phú Yên	0106000020E6100...



# Nội dung

---

## ◆ SQL SELECT

### ◆ Truy vấn thuộc tính

### ◆ Bài tập 1

## ◆ SQL quan hệ không gian

### ◆ ST\_Intersects(geometry A, geometry B)

### ◆ ST\_Crosses(geometry A, geometry B)

### ◆ ST\_Overlaps(geometry A, geometry B)

### ◆ ST\_Touches(geometry A, geometry B)

### ◆ ST\_Within(geometry A, geometry B)

### ◆ ST\_Contains(geometry A, geometry B)

### ◆ ST\_Distance(geometry A, geometry B)

### ◆ ST\_DWithin(geometry A, geometry B, distance R)

### ◆ Bài tập 2

## ◆ SQL phép kết không gian

### ◆ JOIN ... ON ...

## ◆ SQL tạo dữ liệu không gian

### ◆ ST\_Buffer(geometry, distance)

### ◆ ST\_Centroid(geometry)

### ◆ ST\_PointOnSurface(geometry)

### ◆ ST\_Union(geometry A, geometry B)

### ◆ ST\_Union(geometry)

### ◆ ST\_Intersection(geometry A, geometry B)

### ◆ Bài tập 3

## ◆ SQL chỉ mục không gian

### ◆ CREATE INDEX ...

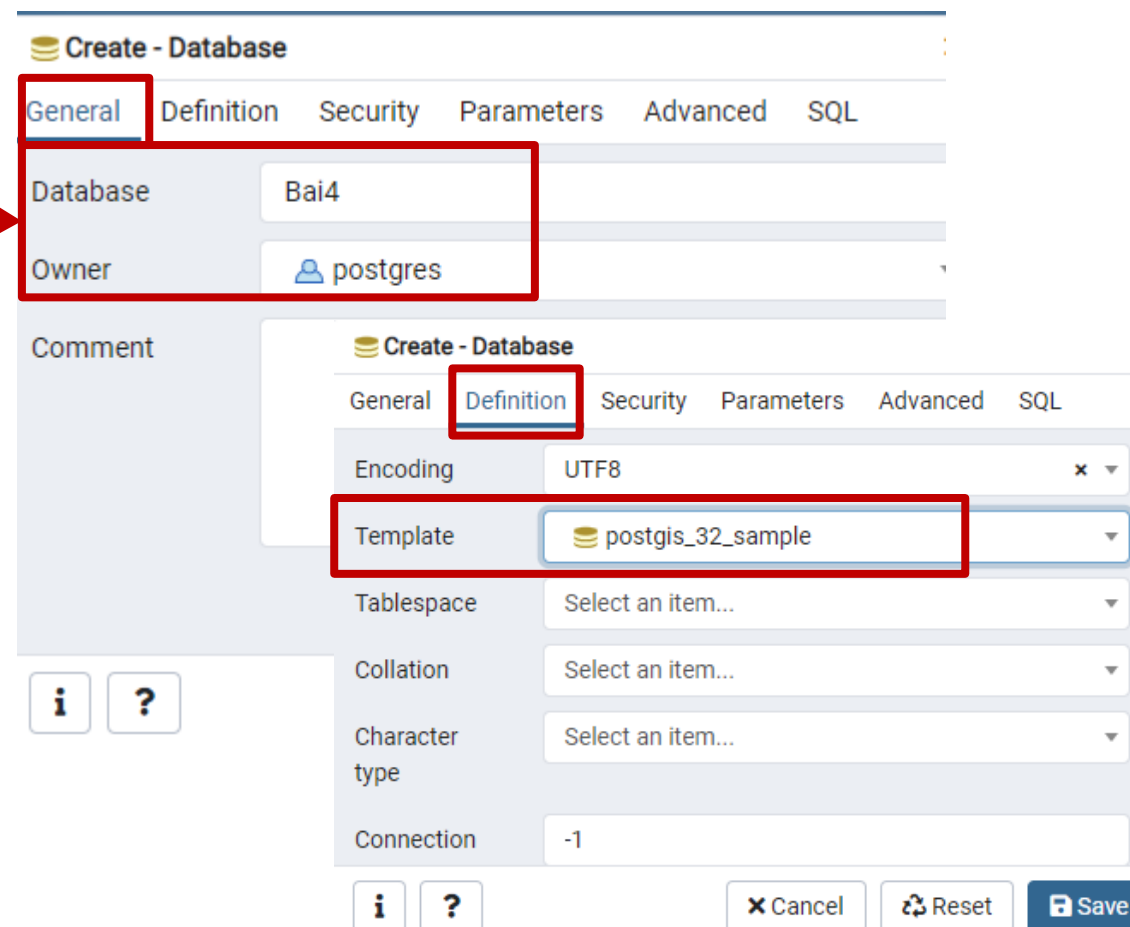
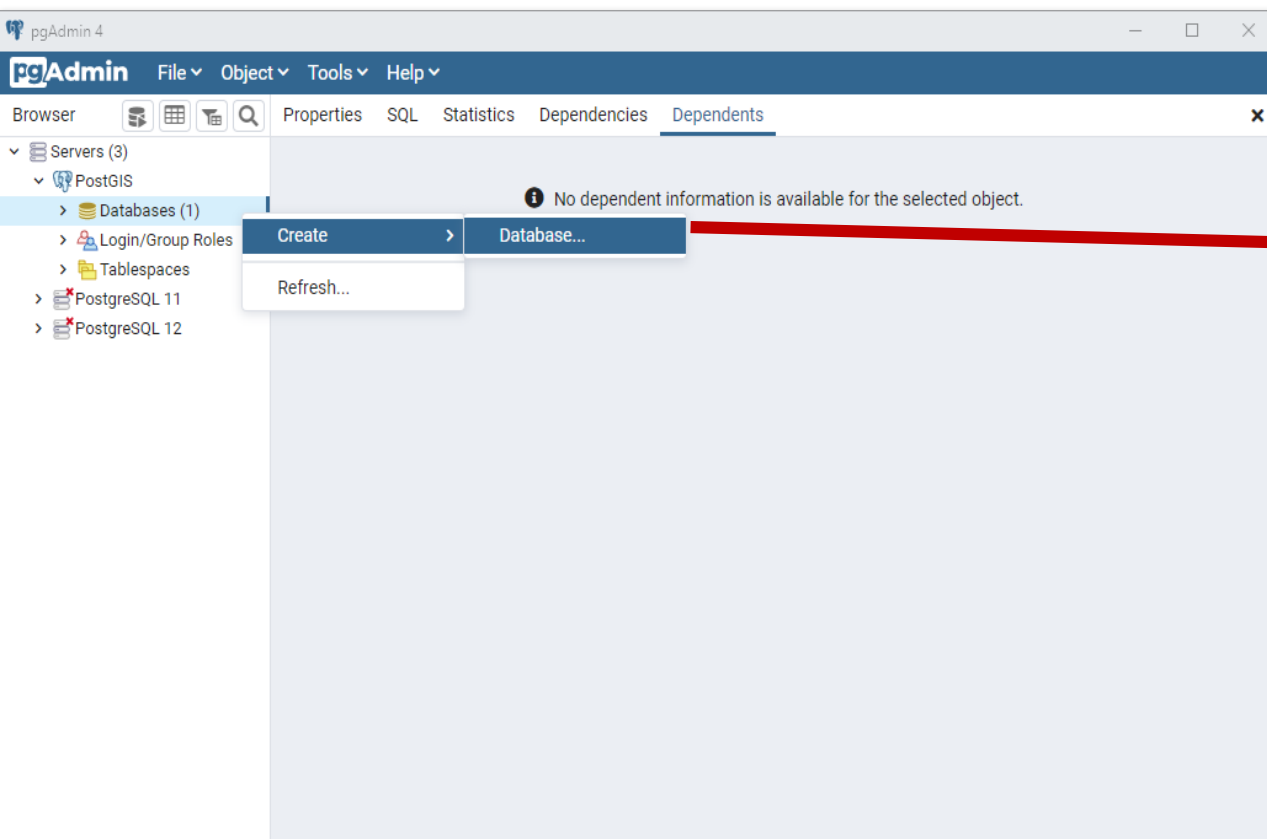
### ◆ DROP INDEX ... ON ... USING GIST ...

# Tạo CSDL địa lý

## ◆ Create database

◆ General: Database = Bai4, Owner = postgres

◆ Definition: **Template = postgis\_32\_sample**

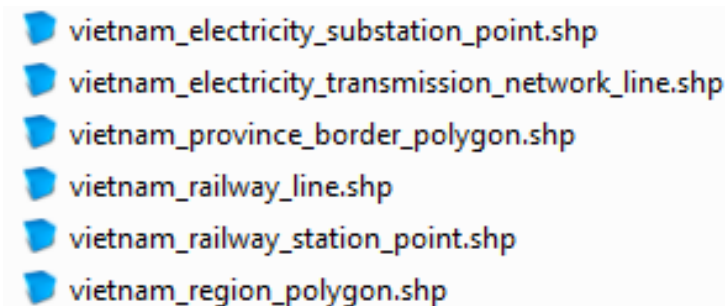
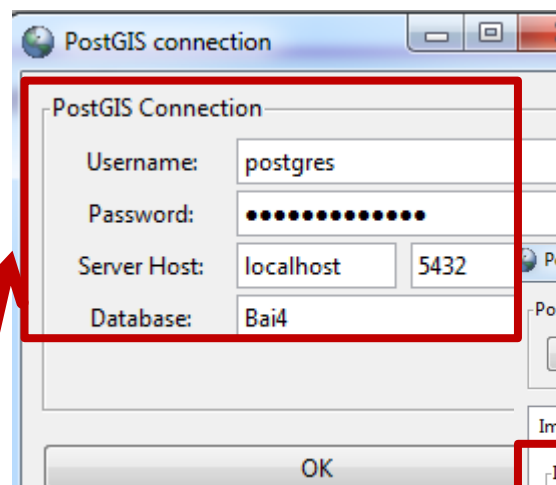
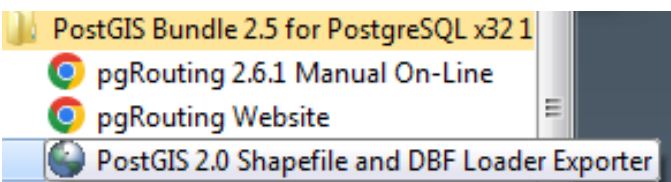


# Thêm dữ liệu địa lý

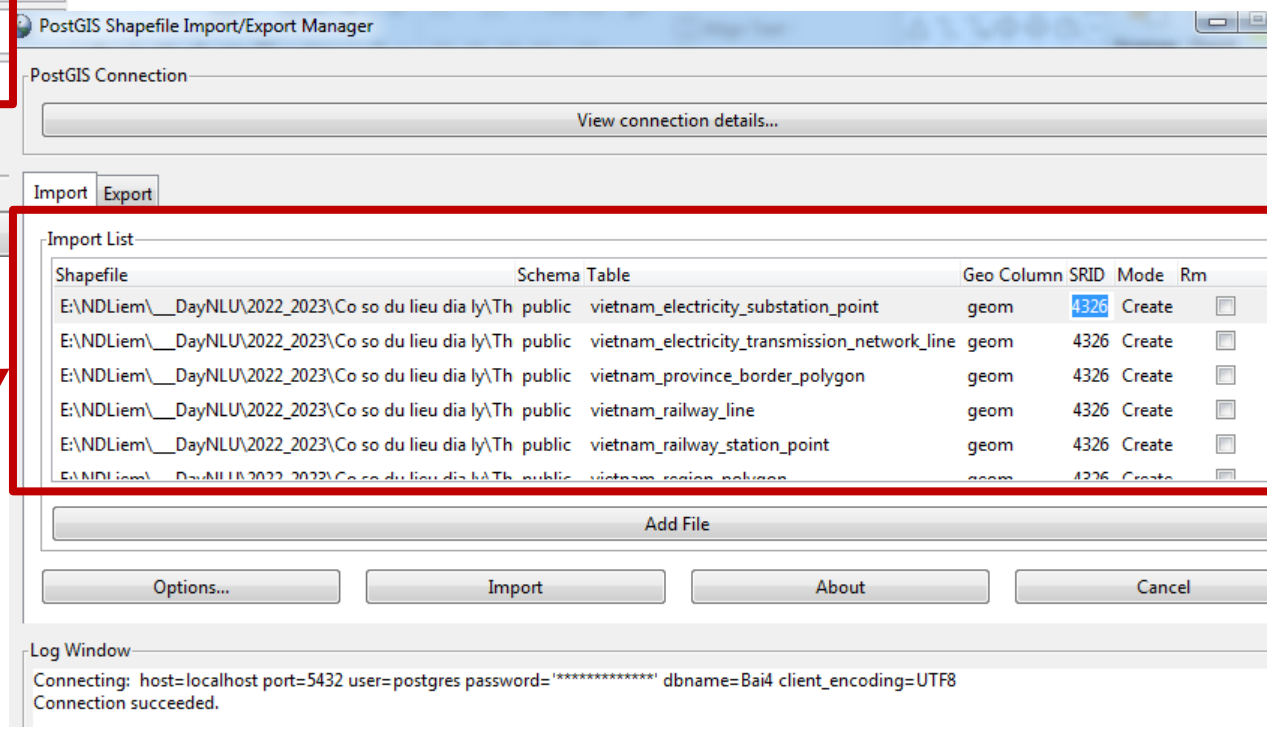
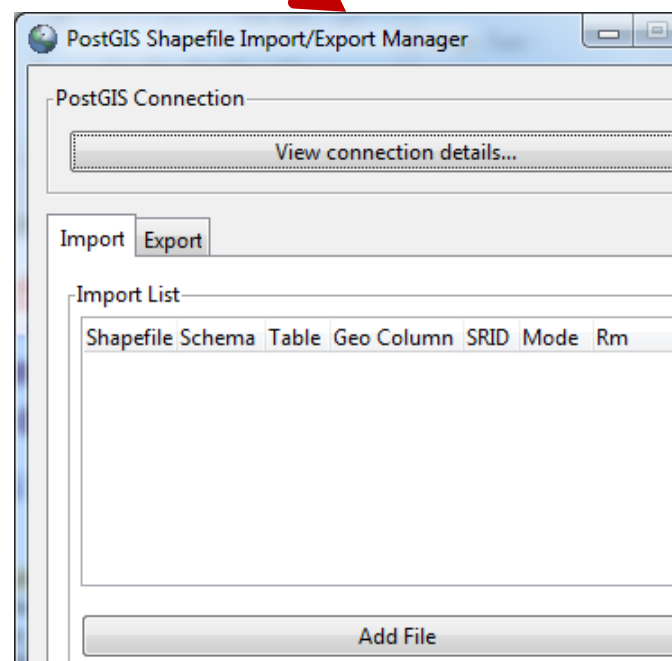
## ◆ Sử dụng PostGIS Shapefile Import/Export Manager

### ◆ Nhập 6 shapefiles vào CSDL địa lý Bai4

### ◆ Khai báo SRID = 4326



Trạm biến áp  
Đường dây tải điện  
Địa giới tỉnh  
Đường sắt  
Ga đường sắt  
Vùng miền

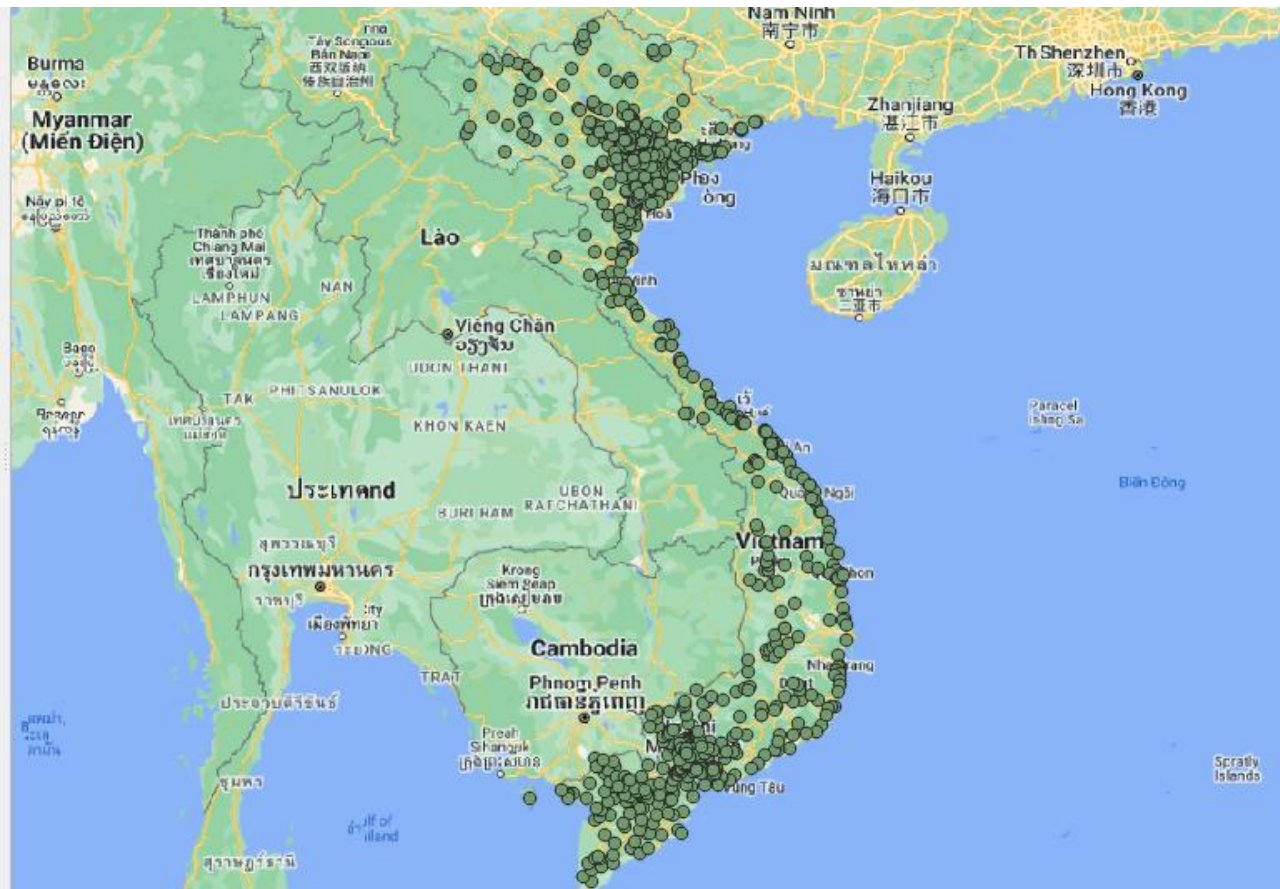




# Mô tả dữ liệu

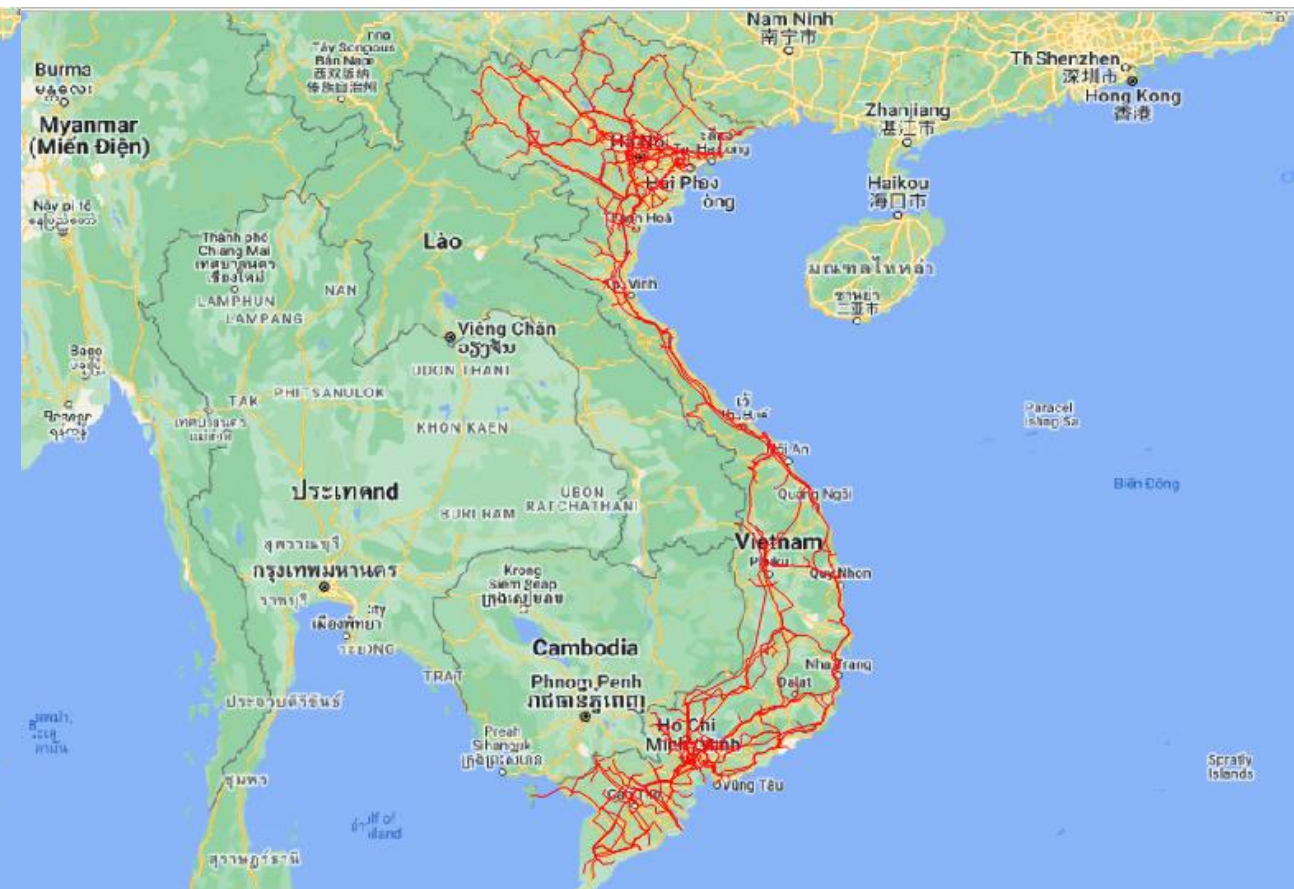
- ◆ Trạm biến áp  
(vietnam\_electricity\_substation\_point)

◆ Voltage: điện áp (kV)



- ◆ Đường dây tải điện  
(vietnam\_electricity\_transmission\_network\_line)

◆ Voltage: điện áp (kV)



# Mô tả dữ liệu

## ◆ Địa giới tỉnh (vietnam\_province\_border\_polygon) ◆ Vùng miền (vietnam\_region\_polygon)

- ◆ Ma: mã tỉnh
- ◆ Ten: tên tỉnh
- ◆ Cap: cấp hành chính
- ◆ DanSo: dân số (người)

- ◆ Ten: tên vùng miền





# Mô tả dữ liệu

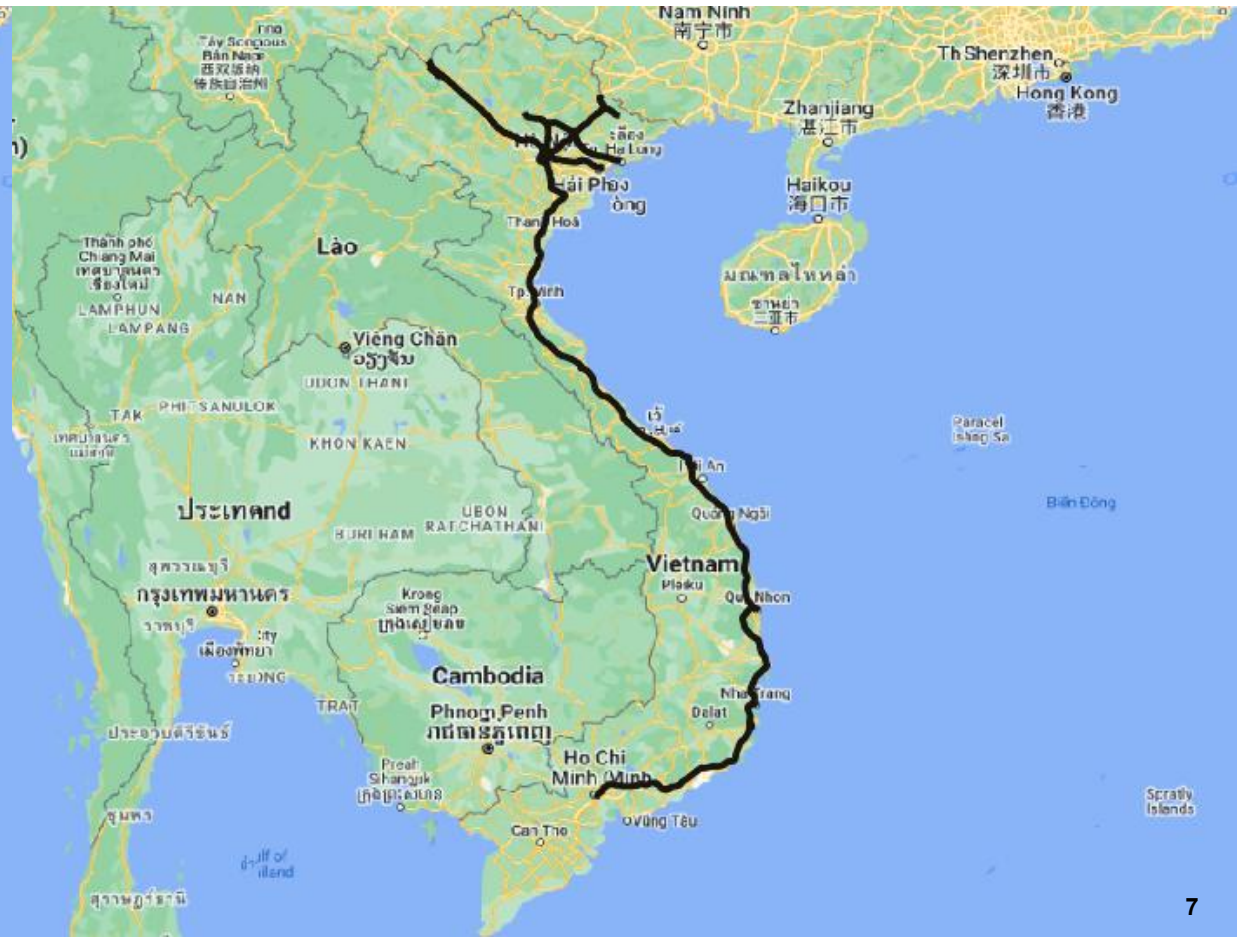
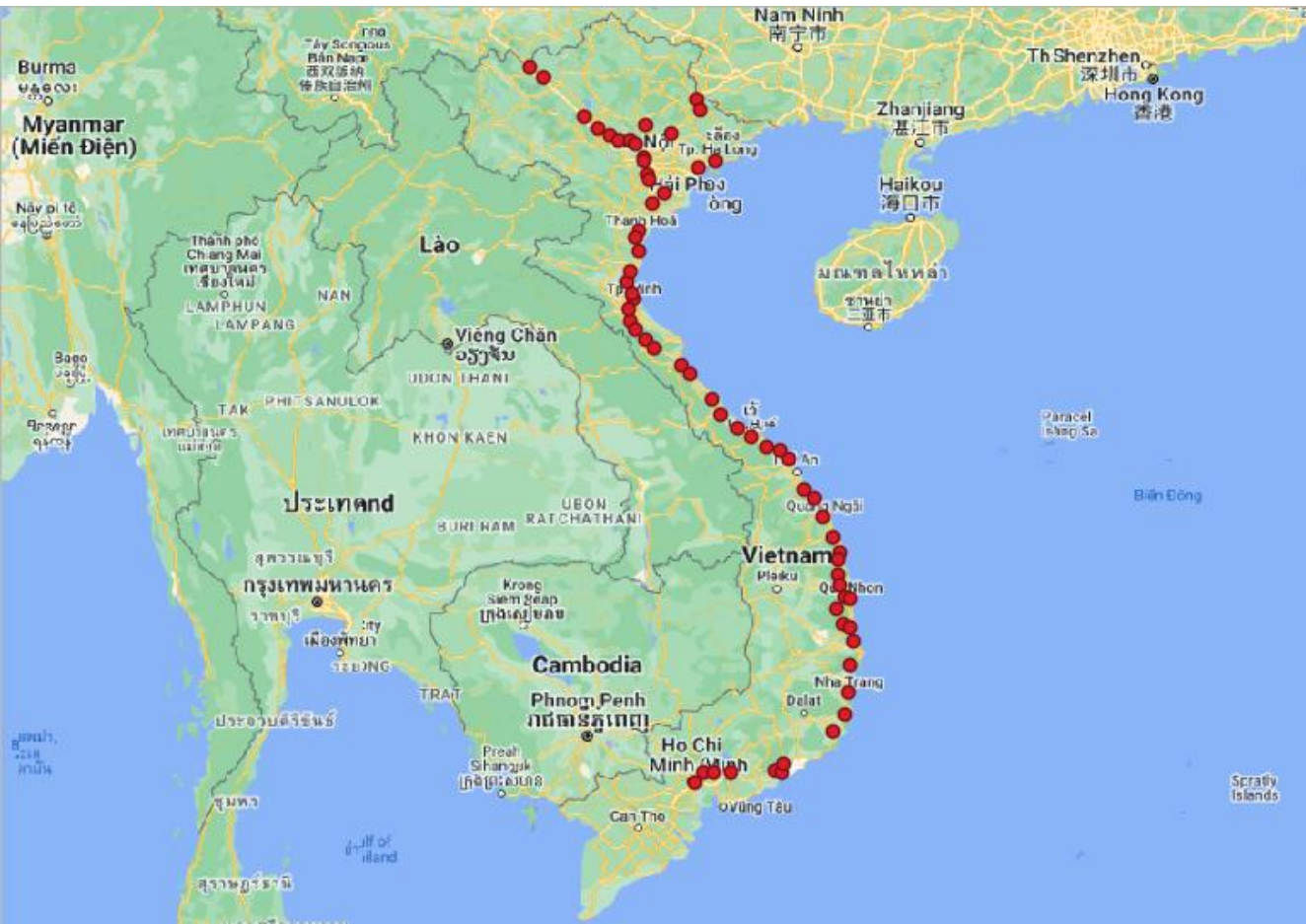
## ◆ Ga đường sắt (vietnam\_railway\_station\_point)

### ◆ Ten\_Ga: tên ga

## ◆ Đường sắt (vietnam\_railway\_line)

### ◆ ten: tên đường

### ◆ cap\_duong: cấp đường



# SQL SELECT – truy vấn thuộc tính

- ◆ SELECT \*
- ◆ FROM vietnam\_province\_border\_polygon
- ◆ WHERE ten LIKE '%Giang%';

Liệt kê thông tin của các tỉnh thành có tên chứa từ Giang trong bảng vietnam\_province\_border\_polygon?



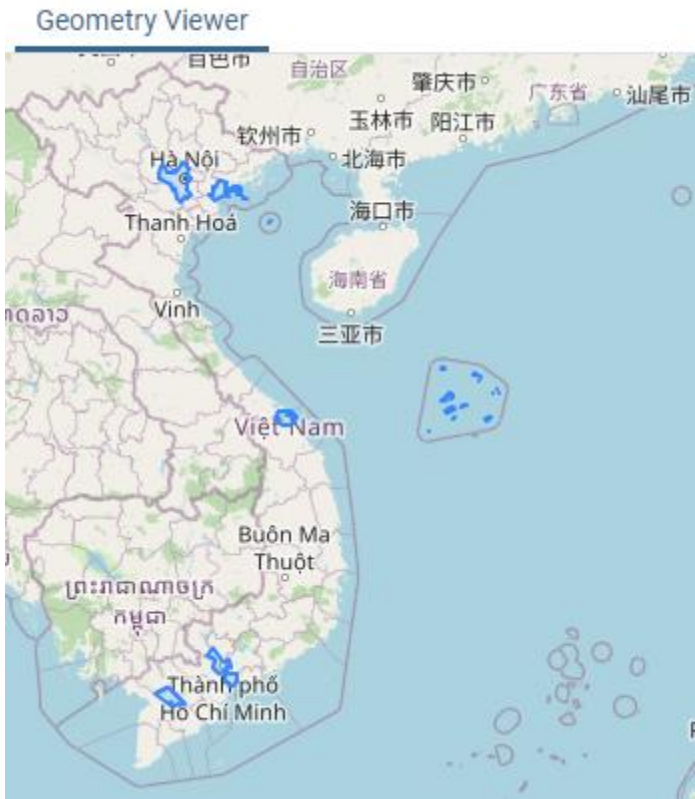
	gid	ma	ten	cap	name	geom
	[PK] integer	character varying (254)	character varying (254)	character varying (254)	character varying (254)	geometry
1	1	24	Tỉnh Bắc Giang	Tỉnh	Bac Giang Province	0106000020E6100...
2	7	02	Tỉnh Hà Giang	Tỉnh	Ha Giang Province	0106000020E6100...
3	12	89	Tỉnh An Giang	Tỉnh	An Giang Province	0106000020E6100...
4	33	82	Tỉnh Tiền Giang	Tỉnh	Tien Giang Province	0106000020E6100...
5	34	91	Tỉnh Kiên Giang	Tỉnh	Kien Giang Province	0106000020E6100...
6	62	93	Tỉnh Hậu Giang	Tỉnh	Hau Giang Province	0106000020E6100...



# SQL SELECT – truy vấn thuộc tính

- ◆ SELECT \*
- ◆ FROM vietnam\_province\_border\_polygon
- ◆ WHERE cap = 'Thành phố Trung ương';

Liệt kê thông tin của các thành phố trực thuộc trung ương trong bảng vietnam\_province\_border\_polygon?

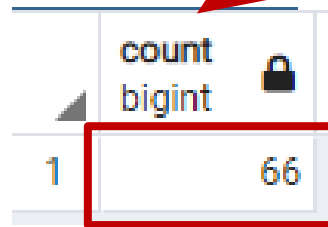


	gid [PK] integer	ma character varying (254)	ten character varying (254)	cap character varying (254)	name character varying (254)	geom geometry
1	2	31	Thành phố Hải Phòng	Thành phố Trung ương	Hai Phong City	0106000020E6100...
2	4	01	Thành phố Hà Nội	Thành phố Trung ương	Ha Noi City	0106000020E6100...
3	18	79	Thành phố Hồ Chí Minh	Thành phố Trung ương	Ho Chi Minh City	0106000020E6100...
4	36	92	Thành phố Cần Thơ	Thành phố Trung ương	Can Tho City	0106000020E6100...
5	43	48	Thành phố Đà Nẵng	Thành phố Trung ương	Da Nang City	0106000020E6100...

# SQL SELECT – truy vấn thuộc tính

- ◆ **SELECT COUNT(\*)**
- ◆ **FROM vietnam\_railway\_station\_point;**

Thống kê số lượng ga đường sắt trong bảng vietnam\_railway\_station\_point?



The screenshot shows a database query result. The first column is labeled 'count' and the second column is labeled 'bigint'. The first row of data shows the value '1' in the first column and '66' in the second column. The value '66' is highlighted with a red box.

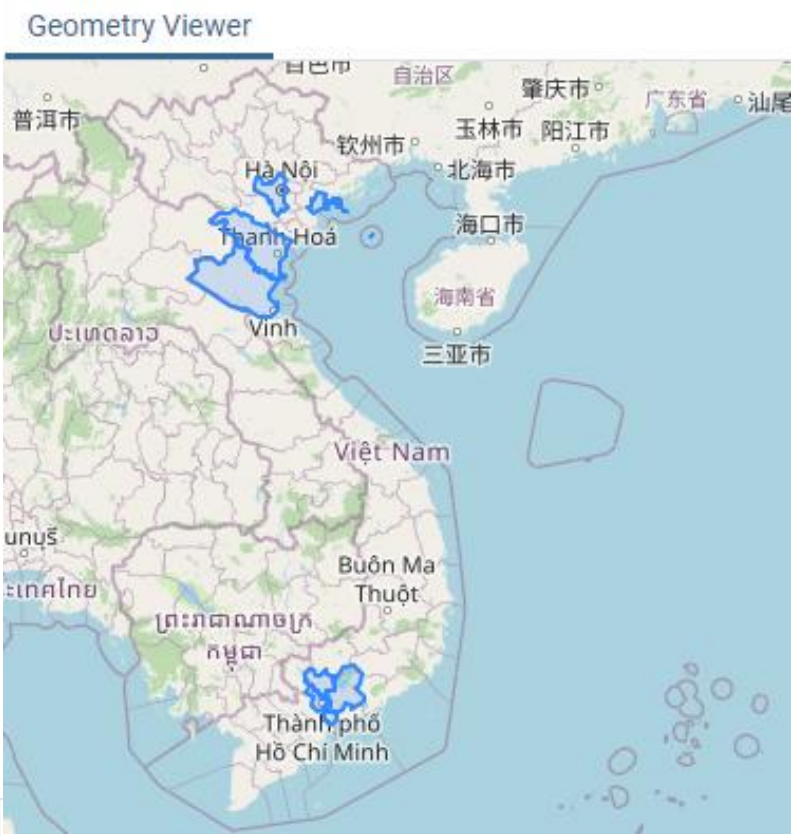
count	bigint
1	66

# SQL SELECT – truy vấn thuộc tính

- ◆ SELECT \*
- ◆ FROM vietnam\_province\_border\_polygon
- ◆ WHERE **DanSo > 2000000**
- ◆ ORDER BY DanSo;

Liệt kê thông tin của các tỉnh thành có dân số trên 2 triệu người trong bảng vietnam\_province\_border\_polygon?

	gid [PK] integer	ma character varying (254)	ten character varying (254)	cap character varying (254)	name character varying (254)	danso double precision	geom geometry
1	2	31	Thành phố Hải Phòng	Thành phố Trung ương	Hai Phong City	2028514	0106000020E6100...
2	30	74	Tỉnh Bình Dương	Tỉnh	Binh Duong Province	2426561	0106000020E6100...
3	20	75	Tỉnh Đồng Nai	Tỉnh	Dong Nai Province	3097107	0106000020E6100...
4	28	40	Tỉnh Nghệ An	Tỉnh	Nghe An Province	3327791	0106000020E6100...
5	8	38	Tỉnh Thanh Hóa	Tỉnh	Thanh Hoa Province	3640128	0106000020E6100...
6	4	01	Thành phố Hà Nội	Thành phố Trung ương	Ha Noi City	8053663	0106000020E6100...
7	18	79	Thành phố Hồ Chí Minh	Thành phố Trung ương	Ho Chi Minh City	8836910	0106000020E6100...



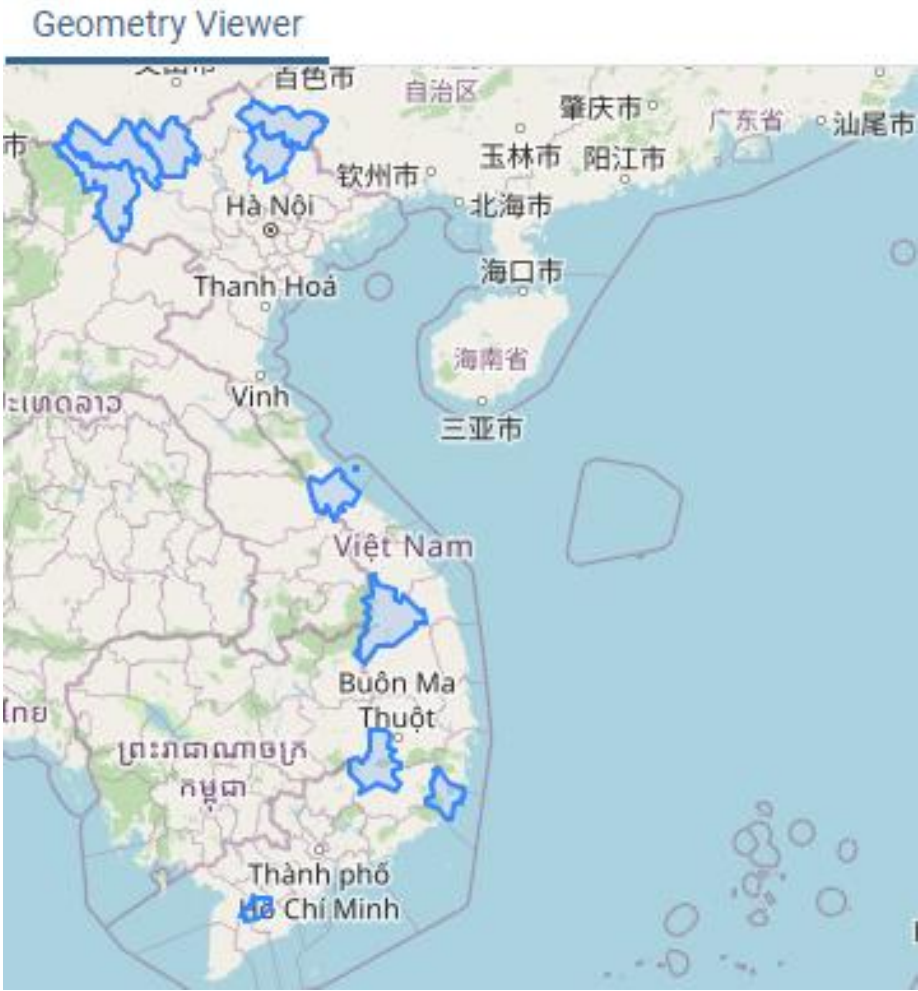


# SQL SELECT – truy vấn thuộc tính

- ◆ SELECT \*
- ◆ FROM vietnam\_province\_border\_polygon
- ◆ ORDER BY DanSo
- ◆ LIMIT 10;

Liệt kê thông tin của top 10 tỉnh thành có dân số ít nhất trong bảng vietnam\_province\_border\_polygon?

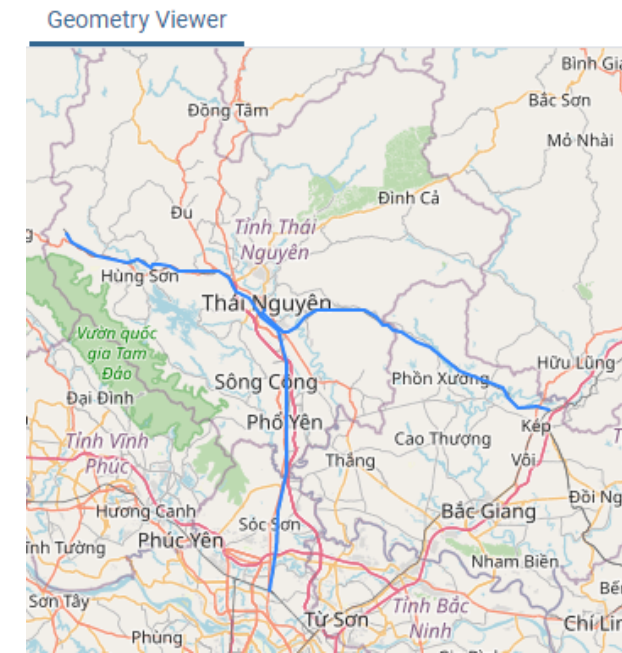
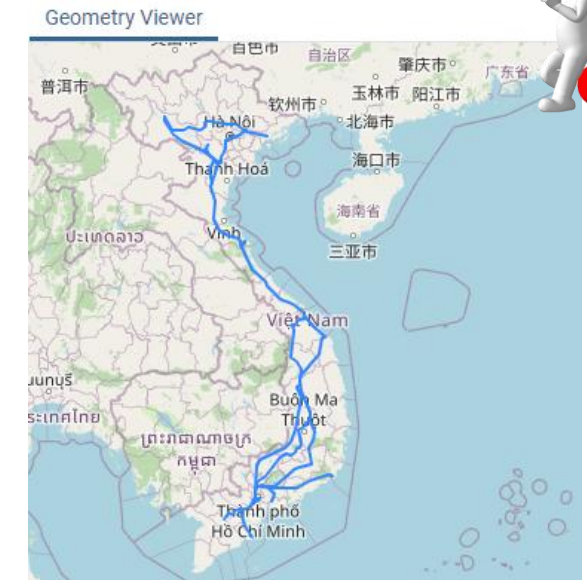
	gid [PK] integer	ma character varying (254)	ten character varying (254)	cap character varying (254)	name character varying (254)	danso double precision	geom geometry
1	46	06	Tỉnh Bắc Kạn	Tỉnh	Bac Kan Province	313905	0106000020E6100...
2	24	12	Tỉnh Lai Châu	Tỉnh	Lai Chau Province	460196	0106000020E6100...
3	23	04	Tỉnh Cao Bằng	Tỉnh	Cao Bang Province	537180	0106000020E6100...
4	61	62	Tỉnh Kon Tum	Tỉnh	Kon Tum Province	540438	0106000020E6100...
5	15	58	Tỉnh Ninh Thuận	Tỉnh	Ninh Thuan Province	590467	0106000020E6100...
6	47	11	Tỉnh Điện Biên	Tỉnh	Dien Bien Province	598856	0106000020E6100...
7	16	67	Tỉnh Đắk Nông	Tỉnh	Dak Nong Province	622168	0106000020E6100...
8	3	45	Tỉnh Quảng Trị	Tỉnh	Quang Tri Province	632375	0106000020E6100...
9	41	10	Tỉnh Lào Cai	Tỉnh	Lao Cai Province	730420	0106000020E6100...
10	62	93	Tỉnh Hậu Giang	Tỉnh	Hau Giang Province	733017	0106000020E6100...



# Bài tập 1: SQL SELECT – truy vấn thuộc tính

1. Liệt kê thông tin của đường dây tải điện 500 kV?

2. Liệt kê thông tin của các tuyến đường sắt đi qua Thái Nguyên?



## SQL quan hệ không gian

---

- ◆ ST\_Intersects(geometry A, geometry B)
- ◆ ST\_Within(geometry A, geometry B)
- ◆ ST\_Contains(geometry A, geometry B)
- ◆ ST\_Distance(geometry A, geometry B)
- ◆ ST\_DWithin(geometry A, geometry B, distance R)
  
- ◆ ST\_Crosses(geometry A, geometry B)
- ◆ ST\_Overlaps(geometry A, geometry B)
- ◆ ST\_Touches(geometry A, geometry B)



Phổ biến

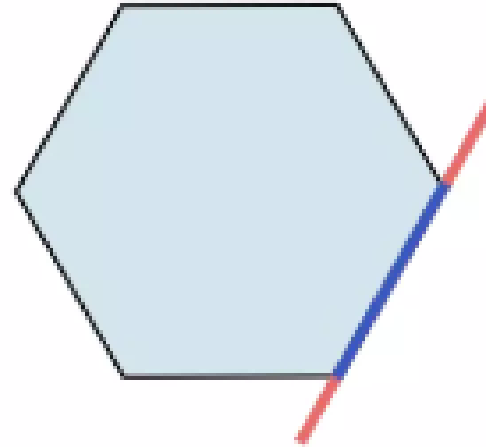
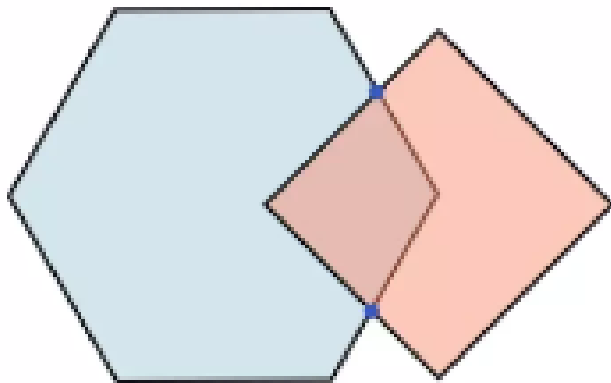
Ít phổ biến



# SQL quan hệ không gian – Intersects

## ◆ ST\_Intersects(geometry A, geometry B)

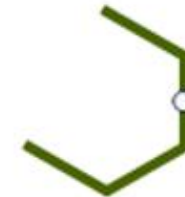
◆ TRUE nếu A, B có ít nhất 1 điểm chung.



Point & Multipoint



Multipoint & Multipoint



Point & Linestring



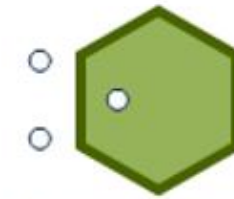
Multipoint & Linestring



Linestring & Linestring



Linestring & Polygon



Multipoint & Polygon



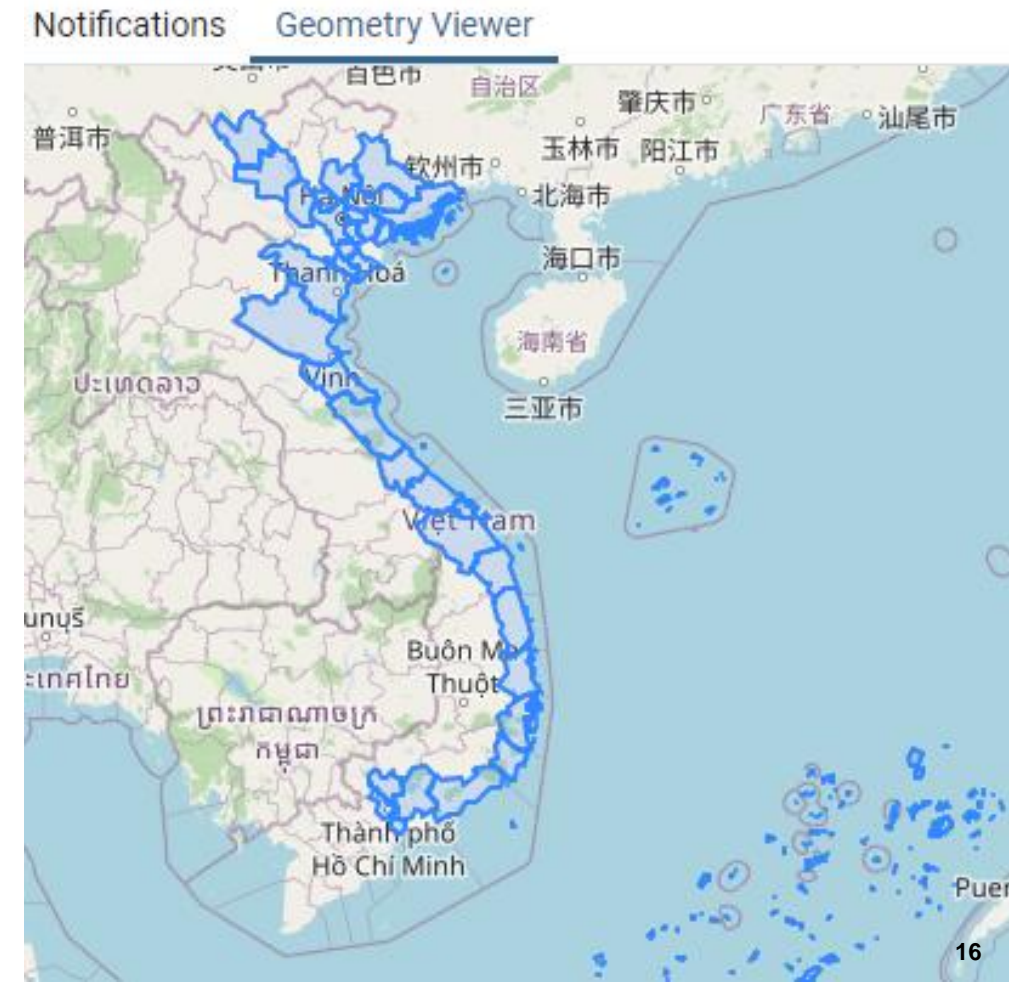
Linestring & Multipolygon

# SQL quan hệ không gian – Intersects

- ◆ SELECT DISTINCT p.Ten, p.geom
- ◆ FROM vietnam\_railway\_line r, vietnam\_province\_border\_polygon p
- ◆ WHERE **ST\_Intersects** (r.geom, p.geom);

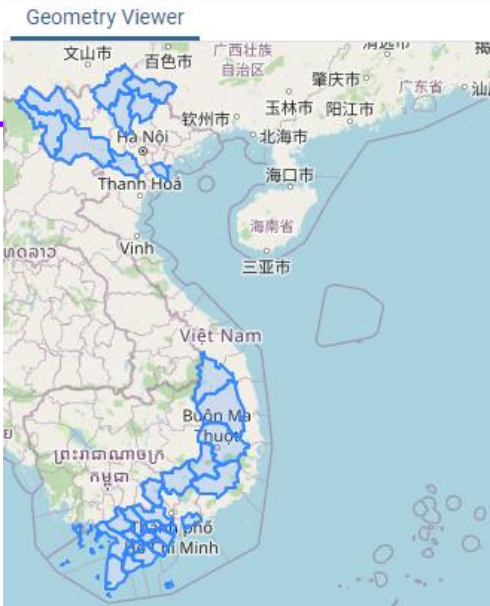
Liệt kê tên, hình học của các tỉnh thành (vietnam\_province\_border\_polygon) giao nhau với đường sắt (vietnam\_railway\_line)?

	ten character varying (254)	geom geometry
1	Tỉnh Đồng Nai	0106000020E6100...
2	Tỉnh Ninh Bình	0106000020E6100...
3	Tỉnh Bình Dương	0106000020E6100...
4	Thành phố Hà Nội	0106000020E6100...
5	Tỉnh Nghệ An	0106000020E6100...
6	Tỉnh Quảng Trị	0106000020E6100...
7	Tỉnh Bình Thuận	0106000020E6100...
8	Tỉnh Yên Bái	0106000020E6100...
9	Tỉnh Thái Nguyên	0106000020E6100...
10	Tỉnh Thanh Hóa	0106000020E6100...



# SQL quan hệ không gian – Intersects

- ◆ SELECT DISTINCT Ten, geom
- ◆ FROM vietnam\_province\_border\_polygon
- ◆ EXCEPT
- ◆ SELECT DISTINCT p.Ten, p.geom
- ◆ FROM vietnam\_railway\_line r, vietnam\_province\_border\_polygon p
- ◆ WHERE **ST\_Intersects**(r.geom, p.geom);



	ten	geom
	character varying (254)	geometry
1	Tỉnh Sơn La	0106000020E6100...
2	Tỉnh Bạc Liêu	0106000020E6100...
3	Tỉnh Trà Vinh	0106000020E6100...
4	Tỉnh An Giang	0106000020E6100...
5	Tỉnh Long An	0106000020E6100...
6	Tỉnh Bến Tre	0106000020E6100...

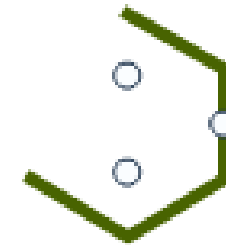
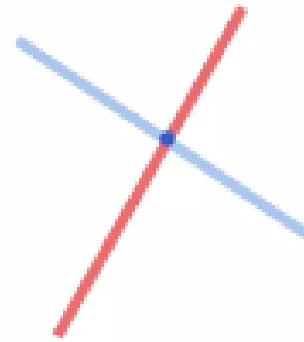
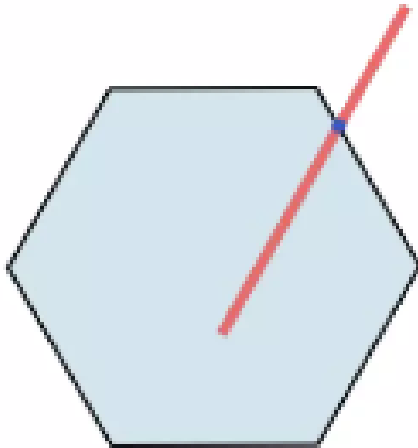
Liệt kê tên, hình học của các tỉnh thành (vietnam\_province\_border\_polygon) không giao nhau với đường sắt (vietnam\_railway\_line)?



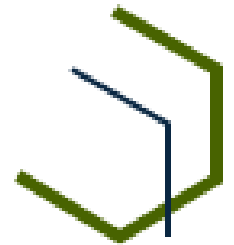
# SQL quan hệ không gian – Crosses

## ◆ ST\_Crosses (geometry A, geometry B)

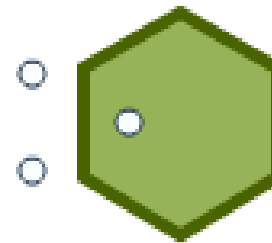
- ◆ TRUE nếu phần giao nhau có số chiều nhỏ hơn số chiều tối đa của A, B và phần giao nhau là phần bên trong của A, B.
- ◆ Thường dùng với lớp dữ liệu dạng đường.



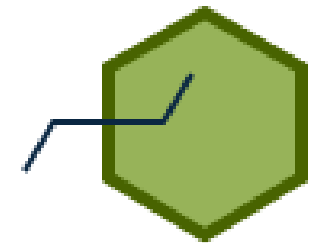
Multipoint & Linestring



Linestring & Linestring



Multipoint & Polygon



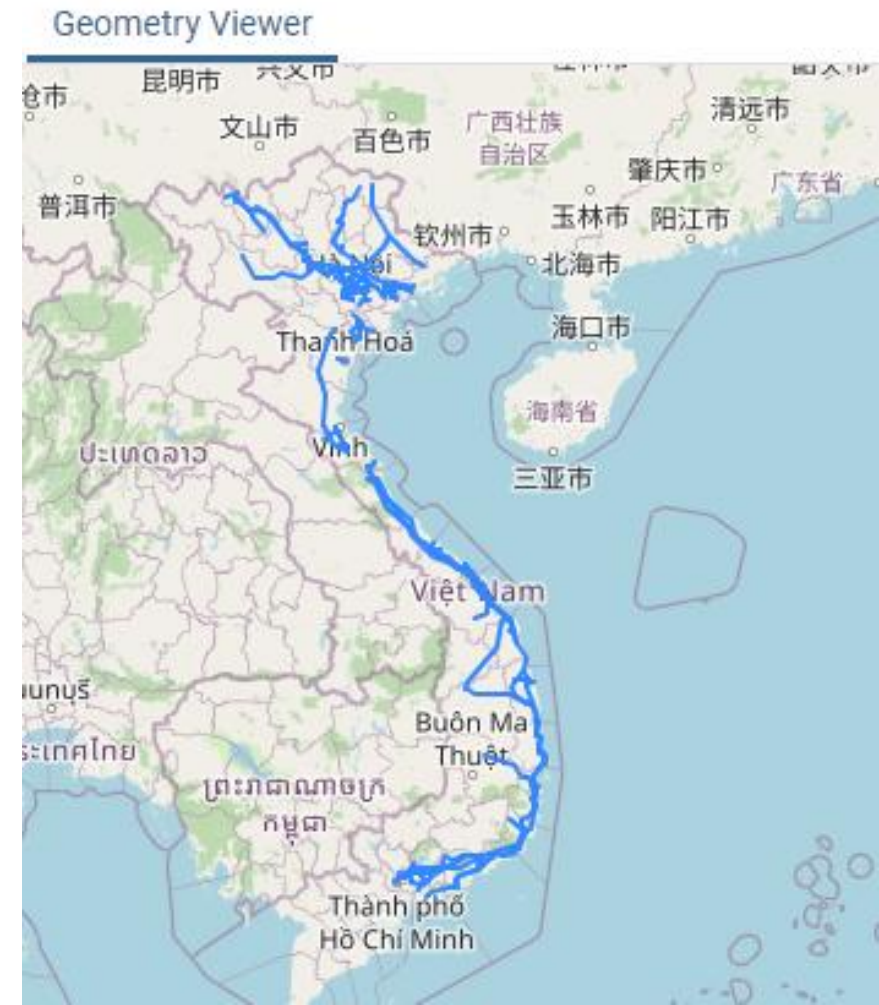
Linestring & Multipolygon

# SQL quan hệ không gian – Crosses

- ◆ SELECT DISTINCT e.Voltage, e.geom
- ◆ FROM vietnam\_railway\_line r, vietnam\_electricity\_transmission\_network\_line e
- ◆ WHERE **ST\_Crosses**(r.geom, e.geom);

Liệt kê điện áp, hình học của các đường dây tải điện (vietnam\_electricity\_transmission\_network\_line) băng qua đường sắt (vietnam\_railway\_line)?

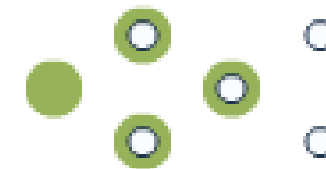
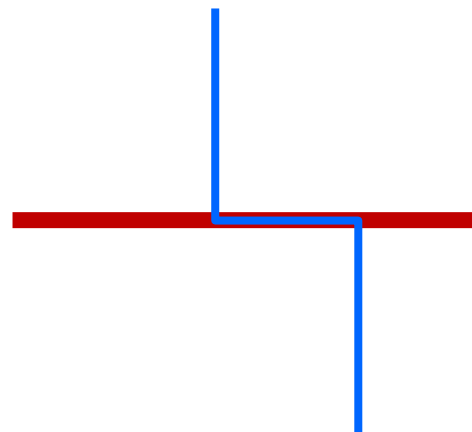
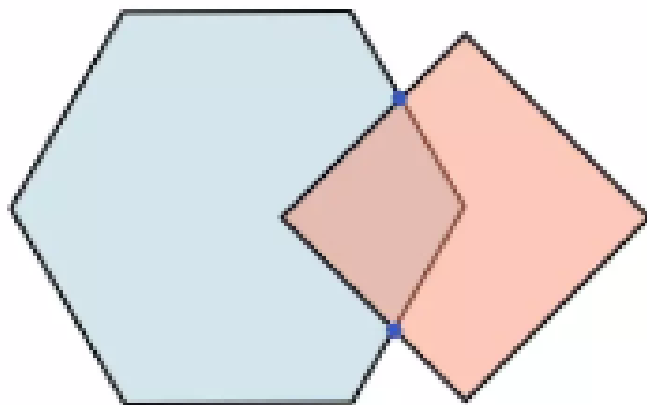
	voltage integer	geom geometry
1	110	0105000020E6100...
2	110	0105000020E6100...
3	110	0105000020E6100...
4	220	0105000020E6100...
5	110	0105000020E6100...
6	500	0105000020E6100...
7	110	0105000020E6100...
8	220	0105000020E6100...
9	110	0105000020E6100...
10	110	0105000020E6100...
11	220	0105000020E6100...



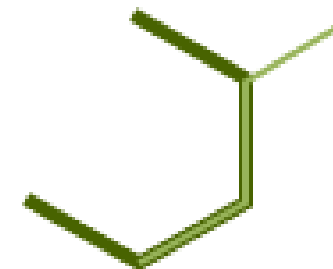
# SQL quan hệ không gian – Overlaps

## ◆ ST\_Overlaps (geometry A, geometry B)

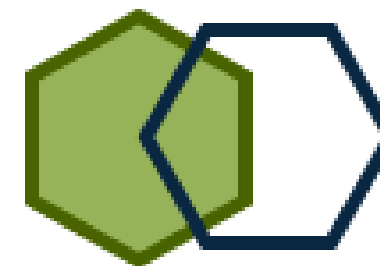
- ◆ TRUE nếu phần giao nhau khác với A, B nhưng có cùng số chiều với A, B.
- ◆ Thường dùng với lớp dữ liệu dạng đường, vùng.



Multipoint & Multipoint



Linestring & Linestring



Polygon & Polygon



## SQL quan hệ không gian – Overlaps

- ◆ **SELECT DISTINCT** e.Voltage, e.geom
- ◆ **FROM** vietnam\_railway\_line r, vietnam\_electricity\_transmission\_network\_line e
- ◆ **WHERE** **ST\_Overlaps**(r.geom, e.geom);

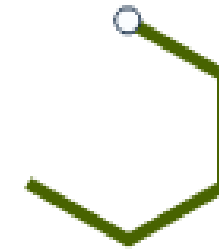
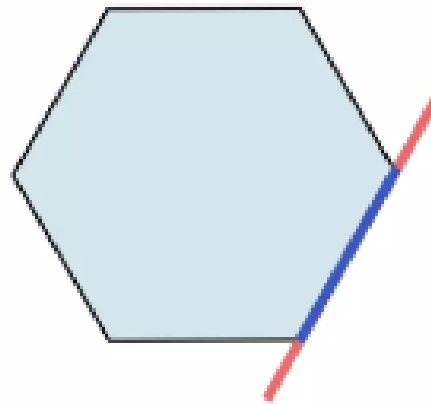
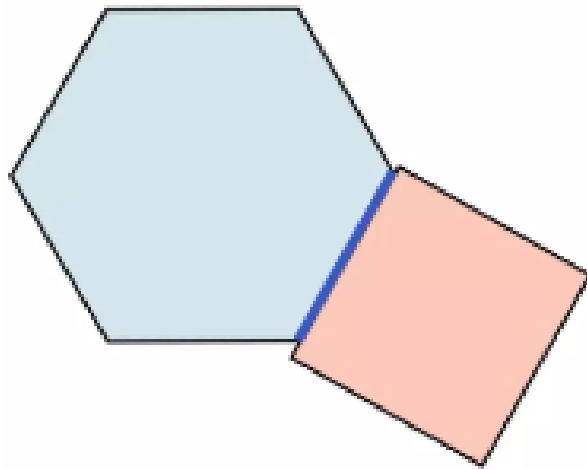
Liệt kê điện áp, hình học của các đường dây tải điện (vietnam\_electricity\_transmission\_network\_line) chồng lấn đường sắt (vietnam\_railway\_line)?

voltage	geom
integer	geometry

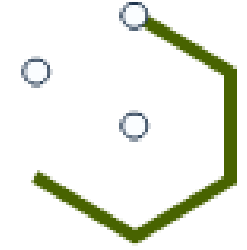
# SQL quan hệ không gian – Touches

## ◆ ST\_Touches (geometry A, geometry B)

- ◆ TRUE nếu phần giao nhau tiếp xúc ở ranh giới của A, B, nhưng không giao nhau ở phần bên trong của A, B.
- ◆ Thường dùng với lớp dữ liệu dạng vùng.



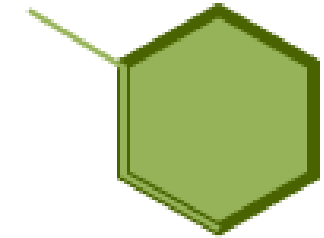
Point & Linestring



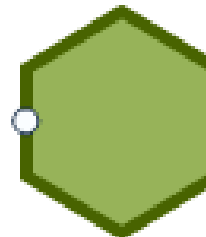
Multipoint & Linestring



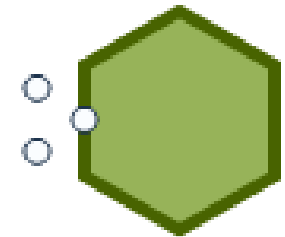
Linestring & Linestring



Linestring & Polygon



Point & Polygon



Multipoint & Polygon

# SQL quan hệ không gian – Touches

- ◆ SELECT ten, geom
- ◆ FROM vietnam\_province\_border\_polygon
- ◆ WHERE **ST\_Touches**((SELECT geom
- ◆ FROM vietnam\_province\_border\_polygon
- ◆ WHERE ten = 'Thành phố Hồ Chí Minh'), geom);

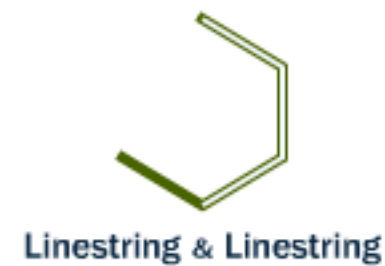
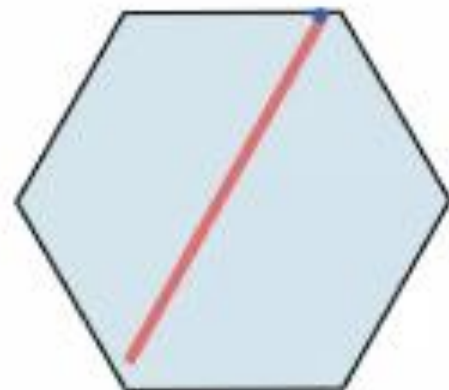
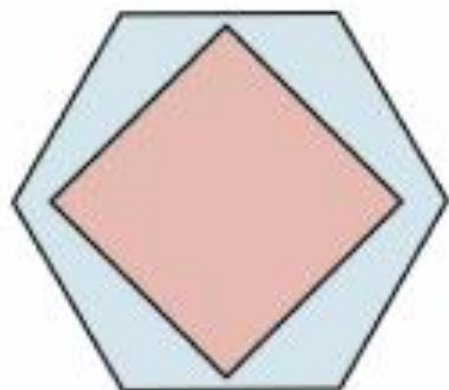
Liệt kê tên, hình học của các tỉnh thành tiếp giáp với Thành phố Hồ Chí Minh (vietnam\_province\_border\_polygon)?

	ten character varying (254)	geom geometry
1	Tỉnh Đồng Nai	0106000020E6100...
2	Tỉnh Long An	0106000020E6100...
3	Tỉnh Bình Dương	0106000020E6100...
4	Tỉnh Tiền Giang	0106000020E6100...
5	Tỉnh Tây Ninh	0106000020E6100...
6	Tỉnh Bà Rịa - Vũng Tàu	0106000020E6100...



# SQL quan hệ không gian – Within, Contains

- ◆ **ST\_Within (geometry A, geometry B)**
  - ◆ TRUE nếu A hoàn toàn nằm trong B.
- ◆ **ST\_Contains (geometry A, geometry B)**
  - ◆ TRUE nếu A hoàn toàn chứa B.





# SQL quan hệ không gian – Within, Contains

- ◆ SELECT ten\_ga, geom
- ◆ FROM vietnam\_railway\_station\_point
- ◆ WHERE **ST\_Within**(geom, (SELECT geom
- ◆ FROM vietnam\_region\_polygon
- ◆ WHERE ten = 'Đông Nam Bộ'));

Liệt kê tên, hình học của các ga đường sắt (vietnam\_railway\_station\_point) nằm trong vùng Đông Nam Bộ (vietnam\_region\_polygon)?

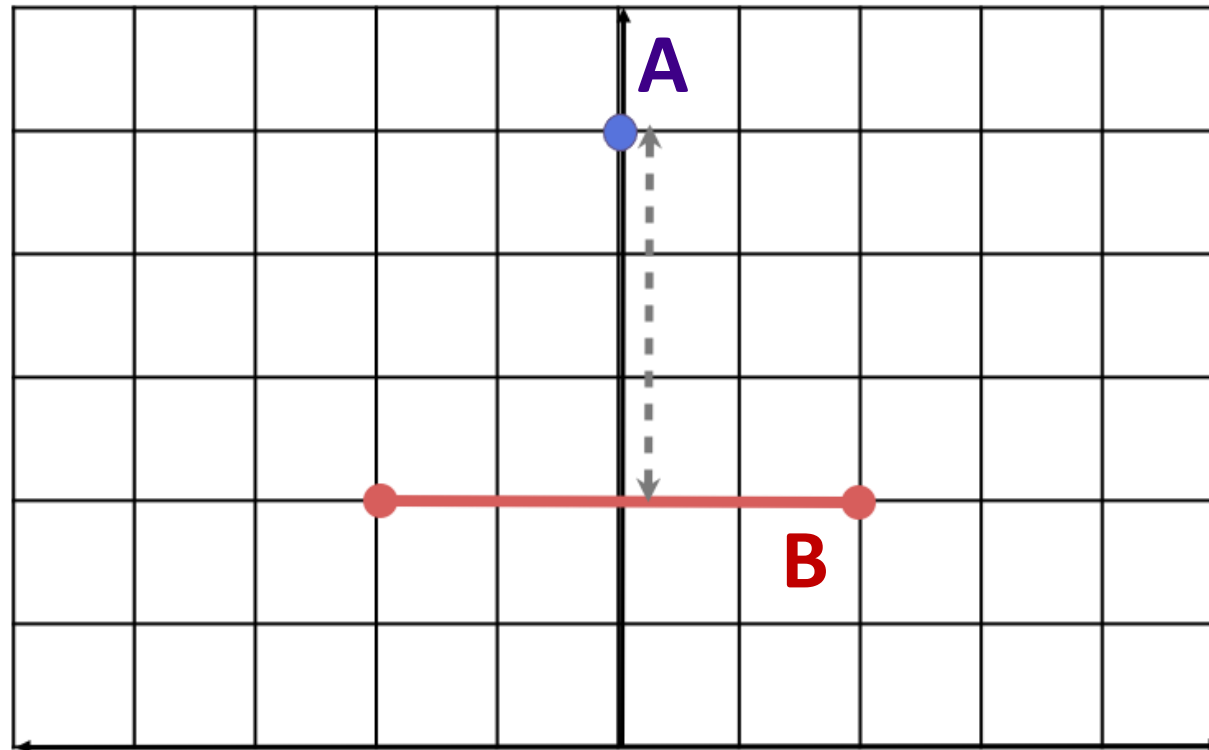
	ten_ga character varying (254)	geom geometry
1	Biên Hòa	0101000020E6100...
2	Sài Gòn	0101000020E6100...
3	Long Khánh	0101000020E6100...
4	Trảng Bom	0101000020E6100...



## SQL quan hệ không gian – Distance

### ◆ ST\_Distance (geometry A, geometry B)

- ◆ Tính toán khoảng cách ngắn nhất giữa A và B (theo hệ tọa độ).



## SQL quan hệ không gian – Distance

---

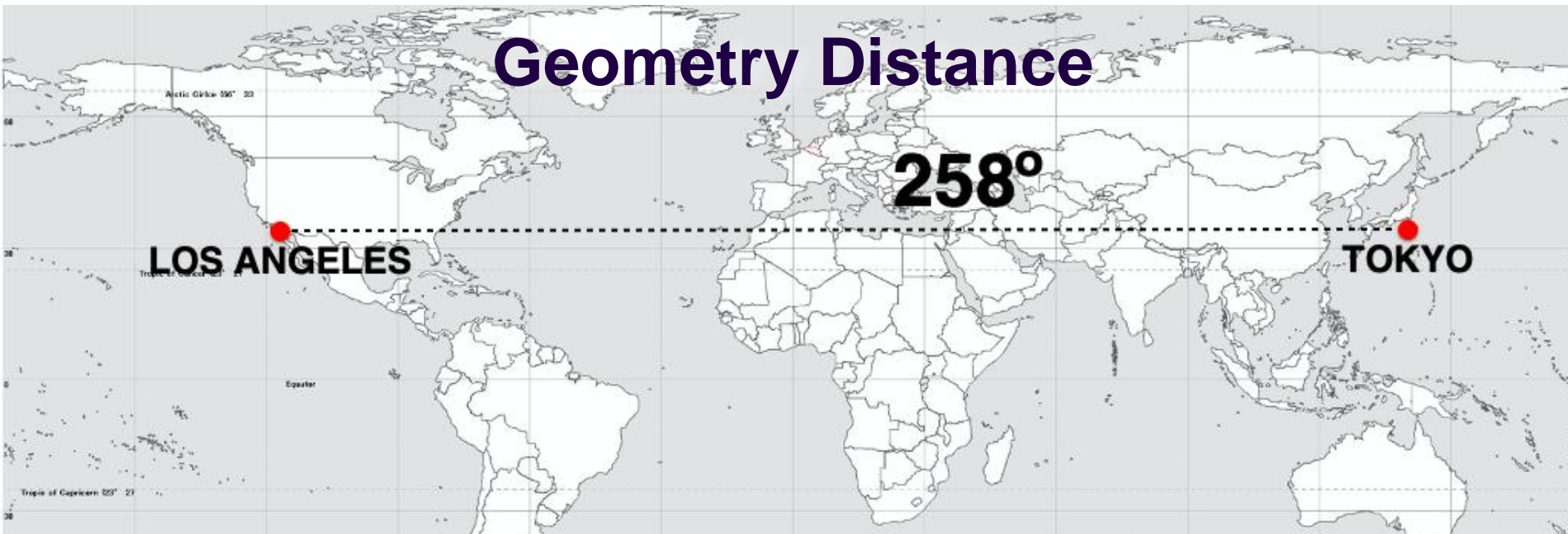
◆ Con đường vòng cung ngắn nhất từ Los Angeles đến Tokyo là gì?

```
SELECT ST_Distance(  
  'SRID=4326;POINT(-118.408 33.943)'::geometry, -- LAX  
  'SRID=4326;POINT( 139.733 35.567)'::geometry) -- NRT  
  AS geometry_distance,  
ST_Distance(  
  'POINT(-118.408 33.943)'::geography, -- LAX  
  'POINT( 139.733 35.567)'::geography) -- NRT  
  AS geography_distance;
```

---

```
geometry_distance:      258.14610835  
geography_distance: 8833973.30246194
```

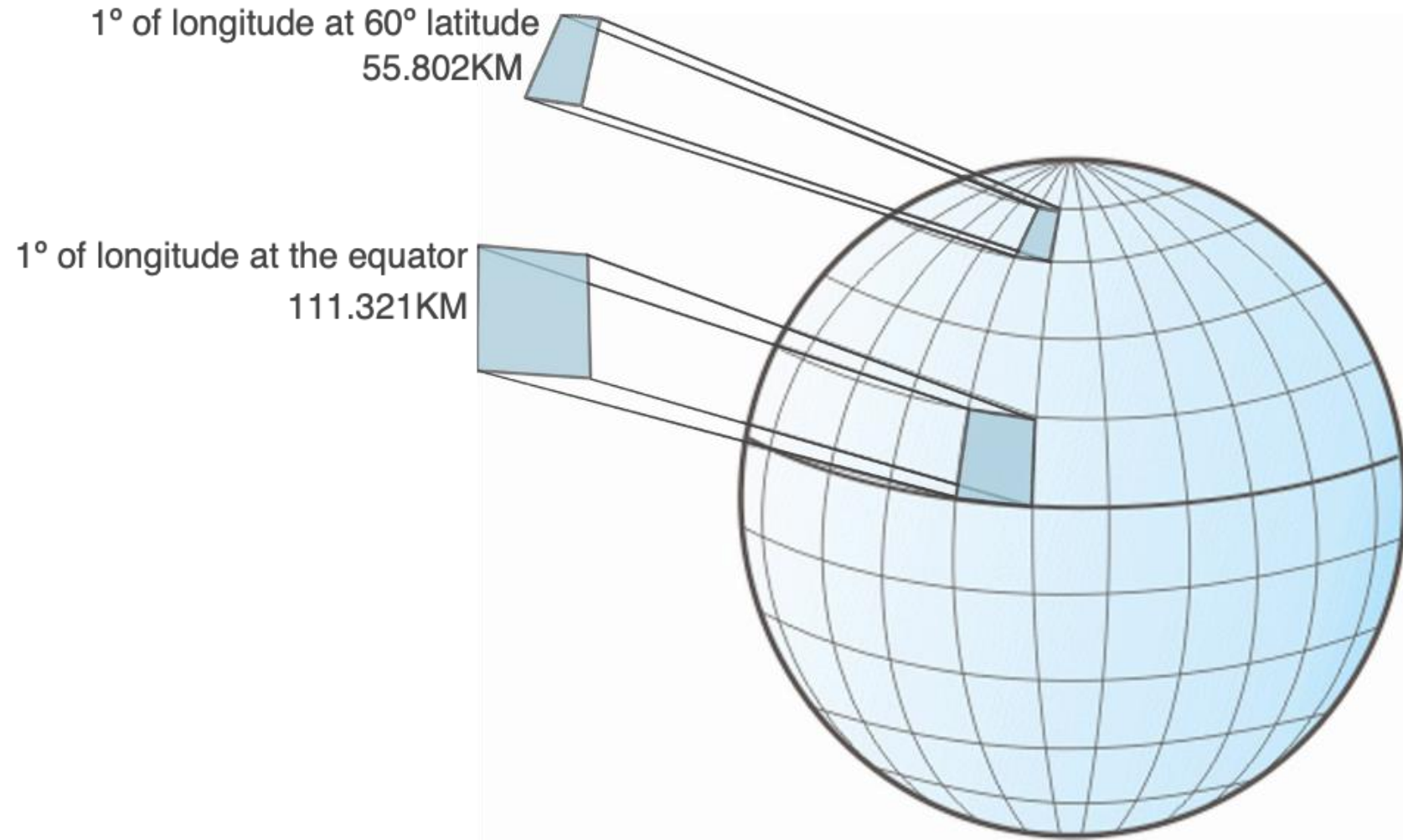
# SQL quan hệ không gian – Distance





## SQL quan hệ không gian – Distance

- ◆ Độ không phải là đơn vị khoảng cách
- ◆ Độ không phải là đơn vị diện tích



## SQL quan hệ không gian – Distance

```
◆ SELECT ST_Distance((SELECT geom::geography
◆ FROM vietnam_province_border_polygon
◆ WHERE ten = 'Thành phố Hồ Chí Minh'),
◆ (SELECT geom::geography
◆ FROM vietnam_province_border_polygon
◆ WHERE ten = 'Thành phố Hà Nội'))/1000;
```

Tính khoảng cách theo km từ Thành phố Hồ Chí Minh đến Thành phố Hà Nội (vietnam\_province\_border\_polygon)?

	?column? double precision
1	1043.20302352165

## SQL quan hệ không gian – DWithin

- ◆ **ST\_DWithin (geometry A, geometry B, distance R)**
  - ◆ TRUE nếu khoảng cách từ A đến B nhỏ hơn R.
  - ◆ Tương đương  $ST\_Distance(geometry A, geometry B) < R$ .



Point & Point (True)



Point & Point (False)



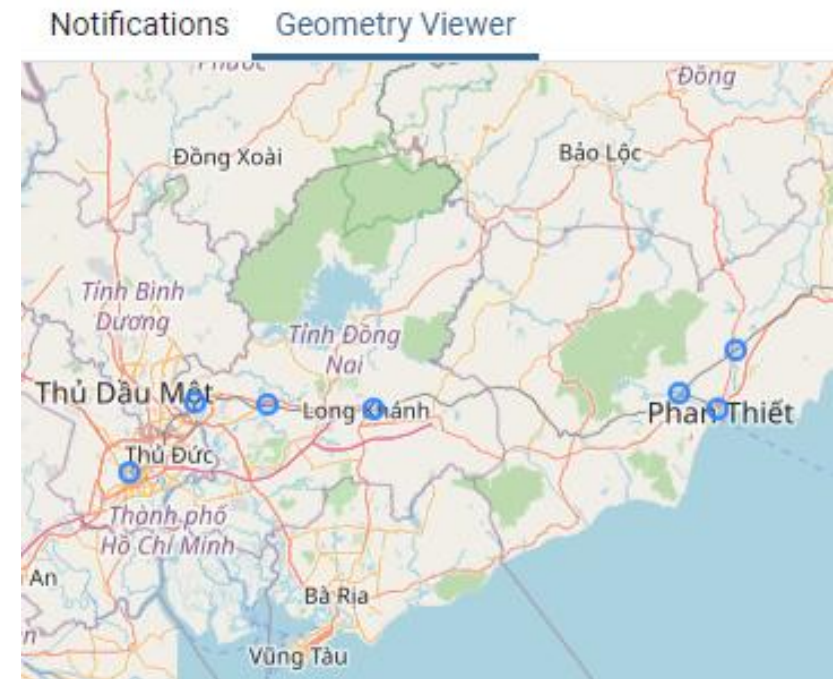
Polygon & Point (True)



Polygon & Point (False)

# SQL quan hệ không gian – DWithin

- ◆ SELECT ten\_ga, geom
- ◆ FROM vietnam\_railway\_station\_point
- ◆ WHERE ST\_DWithin(geom ::geography,
- ◆ (SELECT geom ::geography
- ◆ FROM vietnam\_region\_polygon
- ◆ WHERE ten = 'Đông Nam Bộ'),
- ◆ 100000);



Liệt kê tên, hình học của các ga đường sắt (vietnam\_railway\_station\_point) nằm trong bán kính 100 km tính từ vùng Đông Nam Bộ (vietnam\_region\_polygon)?

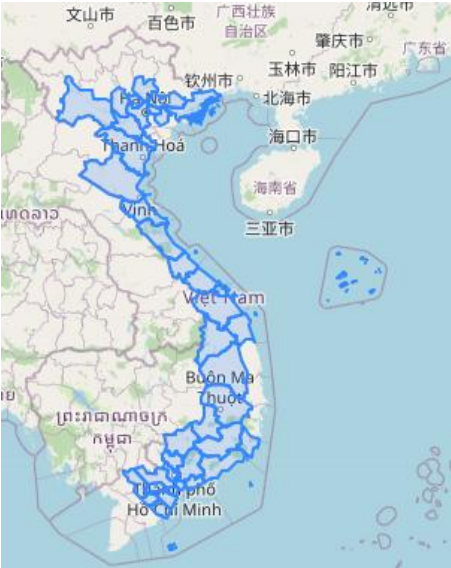
	ten_ga character varying (254)	geom geometry
1	Mương Mán	0101000020E6100...
2	Phan Thiết	0101000020E6100...
3	Biên Hòa	0101000020E6100...
4	Sài Gòn	0101000020E6100...
5	Long Khánh	0101000020E6100...
6	Trảng Bom	0101000020E6100...
7	Ma Lâm	0101000020E6100...



## Bài tập 2: SQL quan hệ không gian

1. Thống kê số ga đường sắt cho từng vùng miền, sắp xếp kết quả theo thứ tự tăng dần của số ga đường sắt?
2. Liệt kê tên của các tỉnh thành (theo thứ tự bảng chữ cái) có đường dây tải điện 500 kV đi qua?
3. Thống kê dân số (người), diện tích (km<sup>2</sup>), mật độ dân số (người/km<sup>2</sup>), hình học của từng tỉnh thành có đường dây tải điện 500 kV đi qua, sắp xếp kết quả theo mật độ dân số giảm dần?

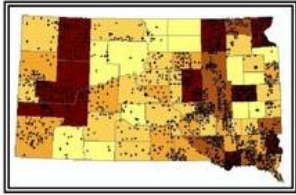
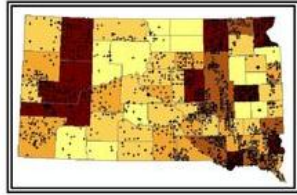
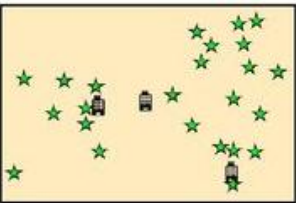

	tenvungmien character varying (254)	soluongga bigint
1	Tây Bắc Bộ	2
2	Đông Nam Bộ	4
3	Đông Bắc Bộ	8
4	Đồng bằng Sông Hồng	9
5	Bắc Trung Bộ	20
6	Duyên hải Nam Trung Bộ	22



	tentinhthanh character varying (254)	danso double precision	dientich double precision	matdods double precision	geom geometry
1	Thành phố Hồ Chí Minh	8836910	2109.1019801095	4189.89223059817	0106000020E6100...
2	Thành phố Hà Nội	8053663	3357.25746358061	2398.88155357932	0106000020E6100...
3	Tỉnh Bắc Ninh	1368840	821.406689822975	1666.45830495367	0106000020E6100...
4	Tỉnh Hải Dương	1892254	1670.46420471318	1132.77135460972	0106000020E6100...
5	Tỉnh Vĩnh Phúc	1151154	1230.0361965318	935.87002012281	0106000020E6100...
6	Tỉnh Bình Dương	2426561	2691.9807917873	901.403534305651	0106000020E6100...
7	Thành phố Đà Nẵng	1134310	1437.27803486519	789.207079273561	0106000020E6100...
8	Tỉnh Tiền Giang	1764185	2405.84899454981	733.289996170403	0106000020E6100...
9	Tỉnh Ninh Bình	982487	1344.09491211501	730.965492945731	0106000020E6100...

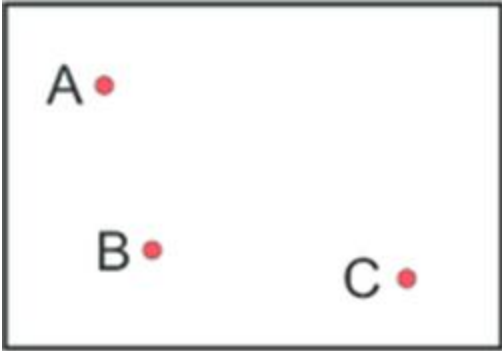
# SQL phép kết không gian

- ◆ **Kết hợp thông tin thuộc tính** từ các bảng khác nhau bằng cách sử dụng các **mối quan hệ không gian** giữa chúng làm khóa liên kết.
- ◆ **Mối quan hệ không gian phổ biến:**
  - ◆ ST\_Intersects(A, B)
  - ◆ ST\_Within(A, B), ST\_Contains(A, B)
  - ◆ ST\_DWithin(A, B, R)
- ◆ **Phân loại phép kết không gian**
  - ◆ Đơn giản: mỗi liên kết 1-1, n-1
  - ◆ Tổng hợp: mỗi liên kết 1-n, n-n
- ◆ **Cú pháp:**
  - ◆ FROM <bảng A> JOIN <bảng B>  
ON <quan hệ không gian A, B>

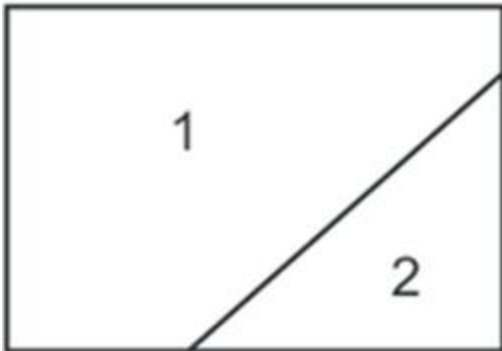
Phép kết Mối quan hệ	Đơn giản	Tổng hợp
ST_Intersects(A, B) ST_Within(A, B), ST_Contains(A, B)	 <b>Trường ← Tỉnh</b> Tên tỉnh của mỗi trường?	 <b>Tỉnh ← Trường</b> Có bao nhiêu trường trong mỗi tỉnh?
ST_DWithin(A, B, R)	 <b>Điểm du lịch ← Khách sạn</b> Điểm du lịch nào nằm gần nhất mỗi khách sạn?	 <b>Khách sạn ← Điểm du lịch</b> Có bao nhiêu điểm du lịch nằm gần nhất mỗi khách sạn?

# SQL phép kết không gian

◆ FROM Điểm X JOIN Vùng Y ON ST\_Intersects(X, Y) → Điểm Z (thuộc tính X, Y)

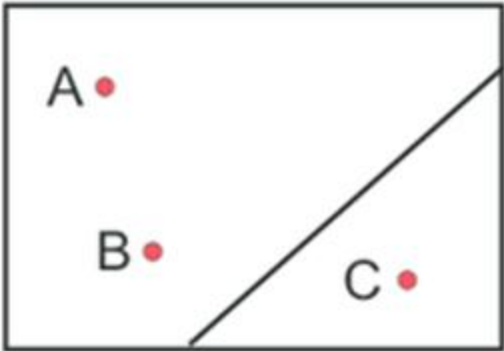


Point	Point Attributes			
A				
B				
C				



Polygon	Polygon Attributes			
1				
2				

Spatial Join



	Point Attributes					Polygon Attributes			
A					1				
B					1				
C					2				

## SQL phép kết không gian

- ◆ **SELECT** s.Ten\_Ga, r.Ten AS TenVungMien
- ◆ **FROM** vietnam\_railway\_station\_point s
- ◆ **JOIN** vietnam\_region\_polygon r
- ◆ **ON** ST\_Within(s.geom, r.geom);

Liệt kê tên vùng miền (vietnam\_region\_polygon) của từng ga đường sắt (vietnam\_railway\_station\_point)?

- ◆ **SELECT** s.Ten\_Ga, r.Ten AS TenVungMien
- ◆ **FROM** vietnam\_railway\_station\_point s,  
vietnam\_region\_polygon r
- ◆ **WHERE** ST\_Within(s.geom, r.geom);

	ten_ga character varying (254)	tenvungmien character varying (254)
1	Chí Chủ	Đông Bắc Bộ
2	Tiên Kiên	Đông Bắc Bộ
3	Việt Trì	Đông Bắc Bộ
4	Hạ Long	Đông Bắc Bộ
5	Đồng Đăng	Đông Bắc Bộ
6	Kép	Đông Bắc Bộ
7	Thái Nguyên	Đông Bắc Bộ
8	Lạng Sơn	Đông Bắc Bộ
9	Vĩnh Yên	Đồng bằng Sông Hồng
10	Phúc Yên	Đồng bằng Sông Hồng
11	Hà Nội	Đồng bằng Sông Hồng
12	Nam Định	Đồng bằng Sông Hồng



## SQL phép kết không gian

- ◆ **SELECT** r.Ten AS TenVungMien, COUNT(s.Ten\_Ga)
- ◆ **FROM** vietnam\_region\_polygon r
- ◆ **JOIN** vietnam\_railway\_station\_point s
- ◆ **ON** ST\_Contains(r.geom, s.geom)
- ◆ **GROUP BY** TenVungMien

Thống kê số lượng ga đường sắt (vietnam\_railway\_station\_point) của từng vùng miền (vietnam\_region\_polygon)?

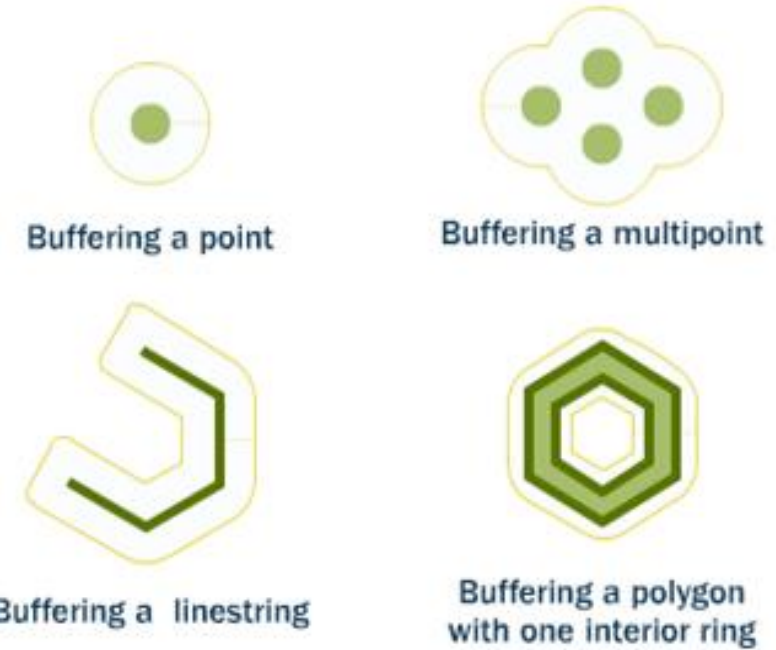
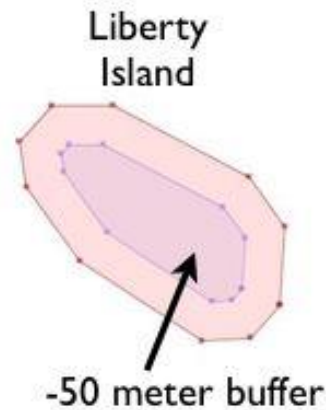
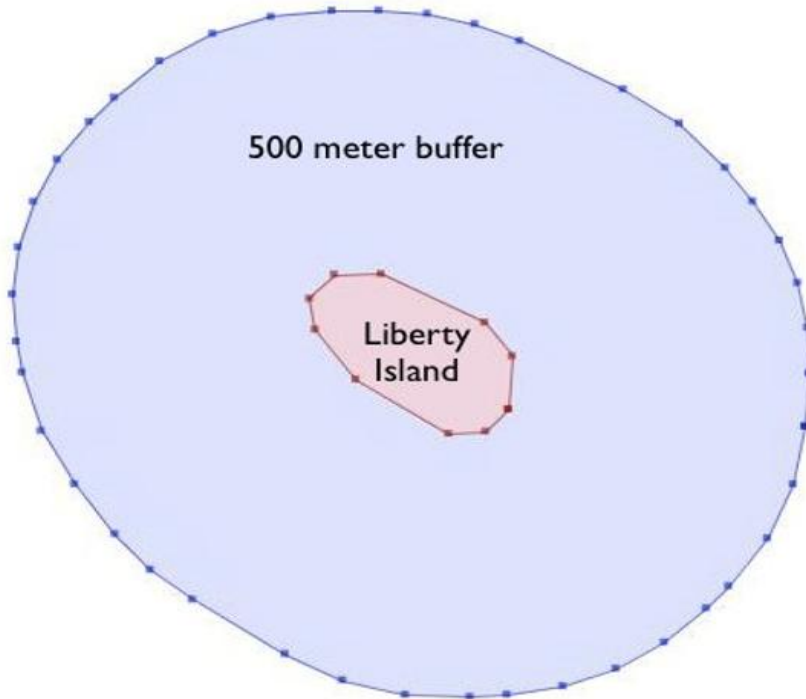
- ◆ **SELECT** r.Ten AS TenVungMien, COUNT(s.Ten\_Ga)
- ◆ **FROM** vietnam\_region\_polygon r, vietnam\_railway\_station\_point s
- ◆ **WHERE** ST\_Contains(r.geom, s.geom)
- ◆ **GROUP BY** TenVungMien

	tenvungmien character varying (254)	count bigint
1	Bắc Trung Bộ	20
2	Đông Bắc Bộ	8
3	Đồng bằng Sông Hồng	9
4	Đông Nam Bộ	4
5	Duyên hải Nam Trung Bộ	22
6	Tây Bắc Bộ	2

# SQL tạo dữ liệu không gian – Buffer

## ◆ ST\_Buffer(geometry, distance)

- ◆ Tạo ra lớp vùng có ranh giới cách lớp đầu vào một khoảng cách cho trước.
- ◆ Khoảng cách có thể âm hoặc dương, với đơn vị theo hệ tọa độ chỉ định.



# SQL tạo dữ liệu không gian – Buffer

- ◆ `SELECT Ten_Ga, ST_Buffer(geom::Geography, 10000) AS geom`
- ◆ `FROM vietnam_railway_station_point;`

Tạo lớp vùng đệm 10 km cho lớp ga đường sắt (vietnam\_railway\_station\_point)?

	ten_ga character varying (254)	geom geography
1	Lào Cai	0103000020E61000...
2	Phổ Lu	0103000020E61000...
3	Yên Bái	0103000020E61000...
4	Chí Chủ	0103000020E61000...
5	Tiên Kiên	0103000020E61000...
6	Việt Trì	0103000020E61000...
7	Vĩnh Yên	0103000020E61000...
8	Phúc Yên	0103000020E61000...
9	Hà Nội	0103000020E61000...



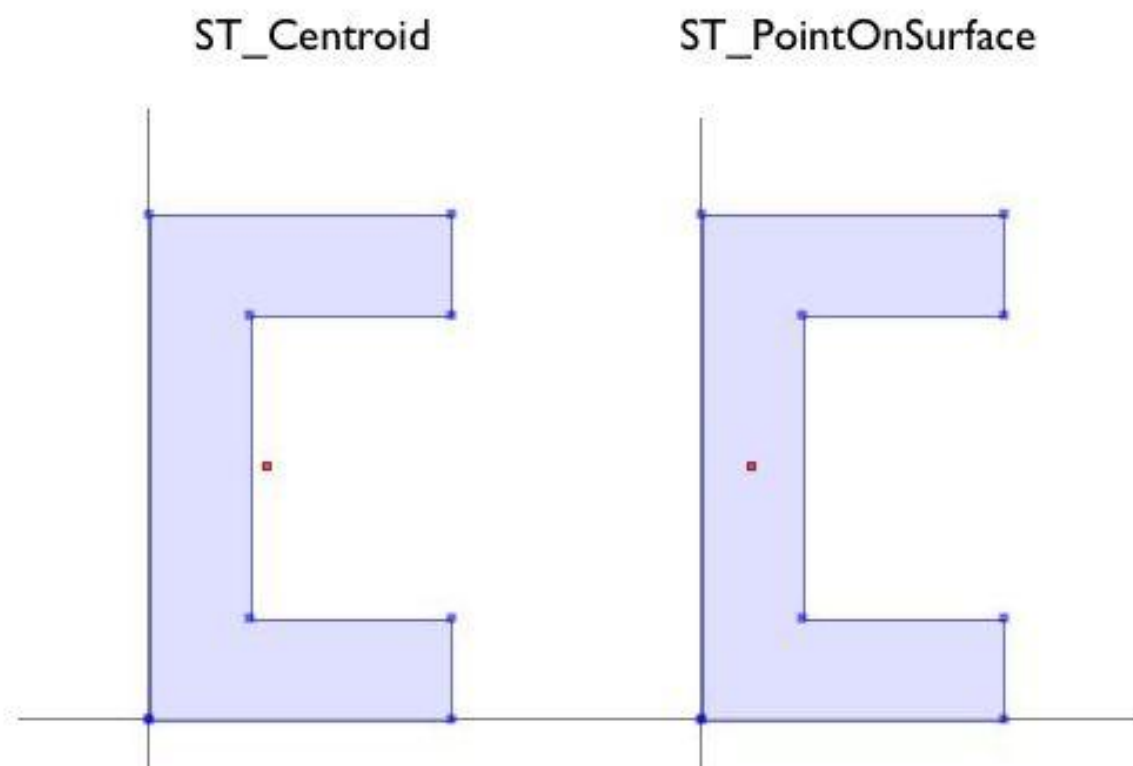
# SQL tạo dữ liệu không gian – Centroid, PointOnSurface

## ◆ ST\_Centroid(geometry)

◆ Trả về một điểm nằm xấp xỉ trọng tâm của lớp đầu vào.

## ◆ ST\_PointOnSurface(geometry)

◆ Trả về một điểm nằm xấp xỉ trọng tâm, nằm trong lớp đầu vào.



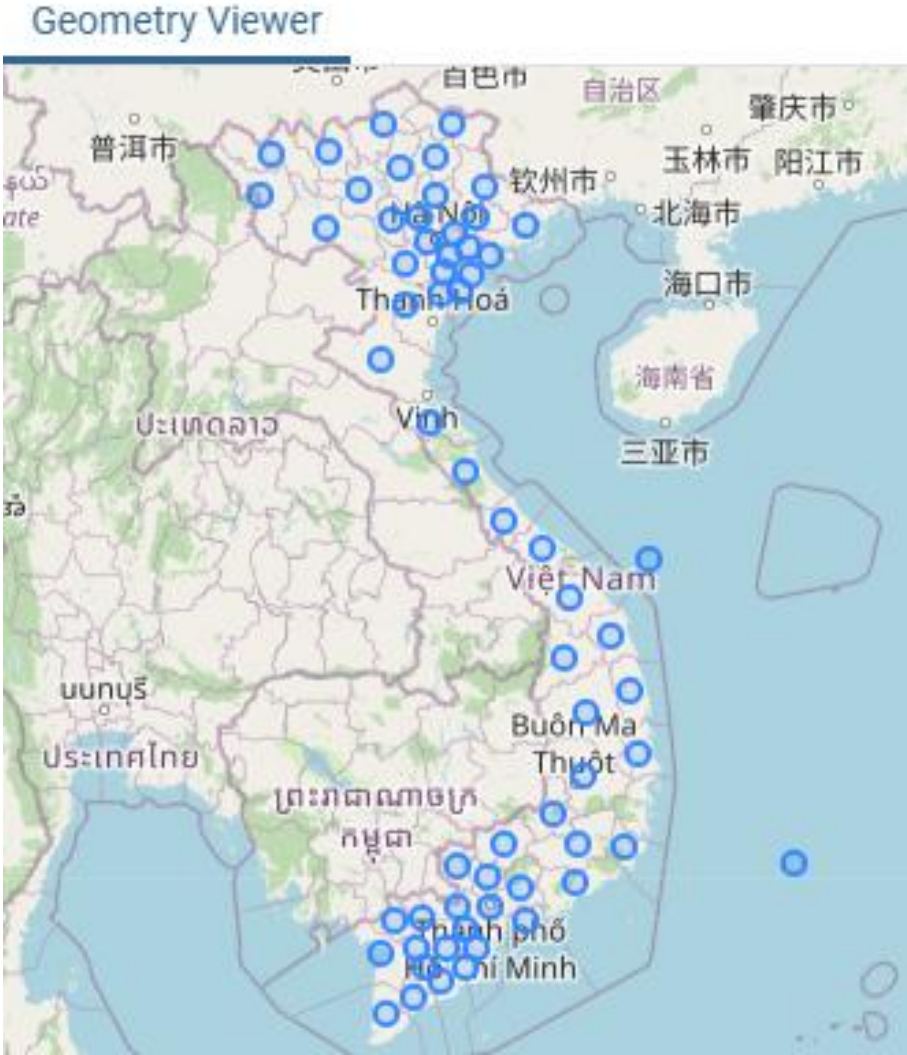


# SQL tạo dữ liệu không gian – Centroid, PointOnSurface

- ◆ **SELECT** Ten, **ST\_X(ST\_Centroid(geom)) AS** KinhDo, **ST\_Y(ST\_Centroid(geom)) AS** ViDo, **ST\_Centroid(geom) AS** geom
- ◆ **FROM** vietnam\_province\_border\_polygon;

Tạo lớp trọng tâm với tên tỉnh thành, kinh độ, vĩ độ cho lớp địa giới tỉnh (vietnam\_province\_border\_polygon)?

	ten character varying (254)	kinhdo double precision	vido double precision	geom geometry
1	Tỉnh Bắc Giang	106.480734627127	21.3574219662858	0101000020E6100...
2	Thành phố Hải Phòng	106.688144770268	20.7921635938652	0101000020E6100...
3	Tỉnh Quảng Trị	106.930156109503	16.7458080821445	0101000020E6100...
4	Thành phố Hà Nội	105.700030001506	20.9989811227515	0101000020E6100...
5	Tỉnh Yên Bái	104.568785204436	21.7767954567135	0101000020E6100...

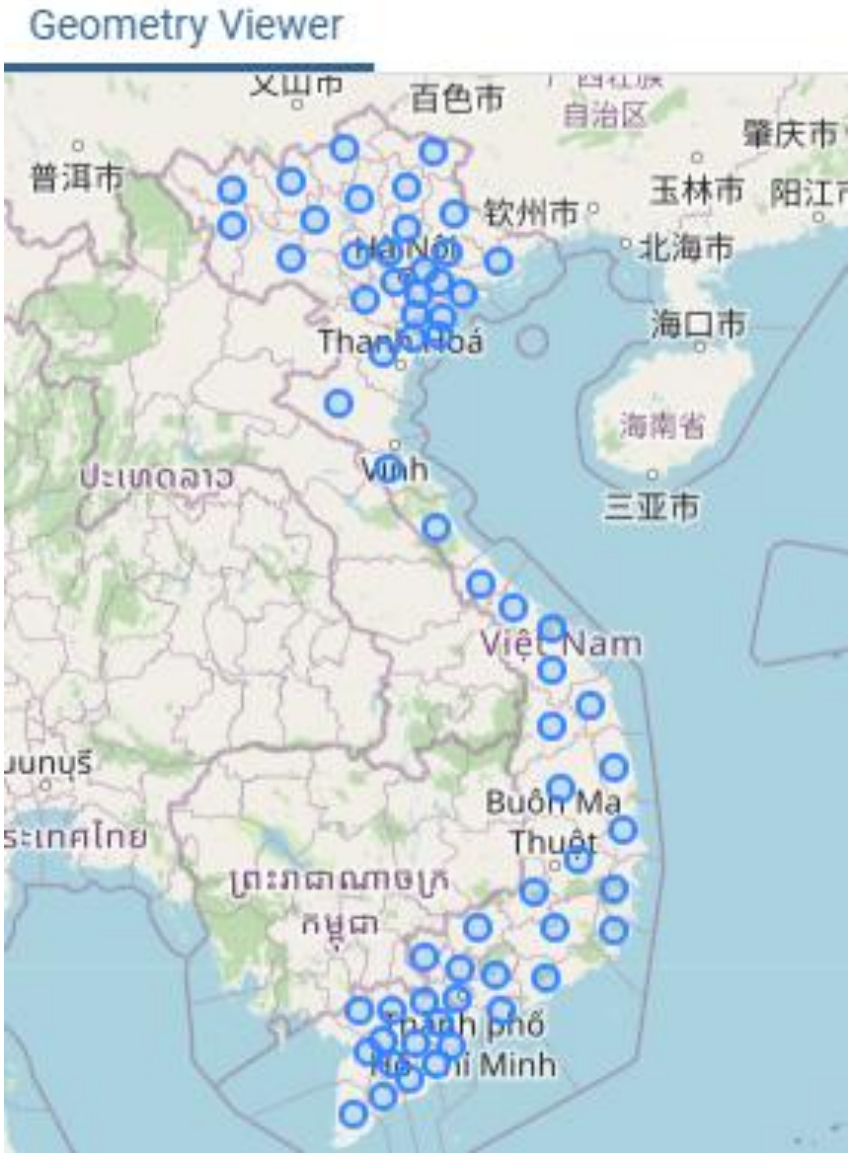


# SQL tạo dữ liệu không gian – Centroid, PointOnSurface

- ◆ `SELECT Ten, ST_X(ST_PointOnSurface(geom)) AS KinhDo, ST_Y(ST_PointOnSurface(geom)) AS ViDo, ST_PointOnSurface(geom) AS geom`
- ◆ `FROM vietnam_province_border_polygon;`

Tạo lớp trọng tâm với tên tỉnh thành, kinh độ, vĩ độ cho lớp địa giới tỉnh (vietnam\_province\_border\_polygon)?

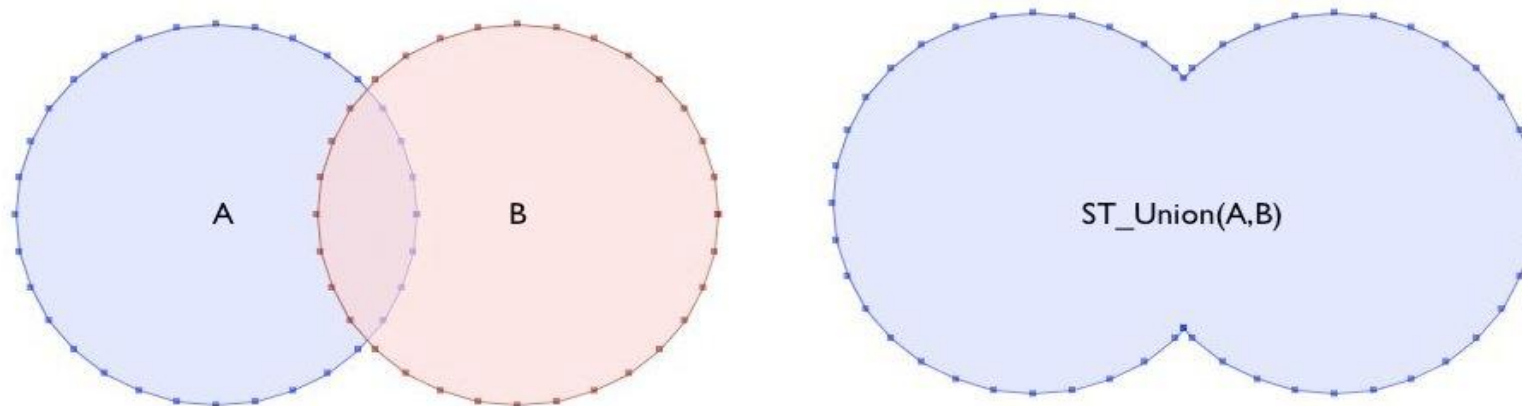
	ten character varying (254)	kinhdo double precision	vido double precision	geom geometry
1	Tỉnh Bắc Giang	106.466270988896	21.37439728	0101000020E6100...
2	Thành phố Hải Phòng	106.70451445895	20.810323715	0101000020E6100...
3	Tỉnh Quảng Trị	106.957646127814	16.72234917	0101000020E6100...
4	Thành phố Hà Nội	105.675881361616	20.97493458	0101000020E6100...
5	Tỉnh Yên Bái	104.51788156783	21.808279995	0101000020E6100...



# SQL tạo dữ liệu không gian – Union

## ◆ ST\_Union(geometry A, geometry B)

◆ Trả về lớp vùng (hoặc đường, điểm) gộp từ hai lớp đầu vào.



## ◆ ST\_Union(geometry)

◆ Trả về lớp hình học đã hợp nhất cho toàn bộ nhóm.

◆ Có thể được sử dụng với câu lệnh SQL GROUP BY.





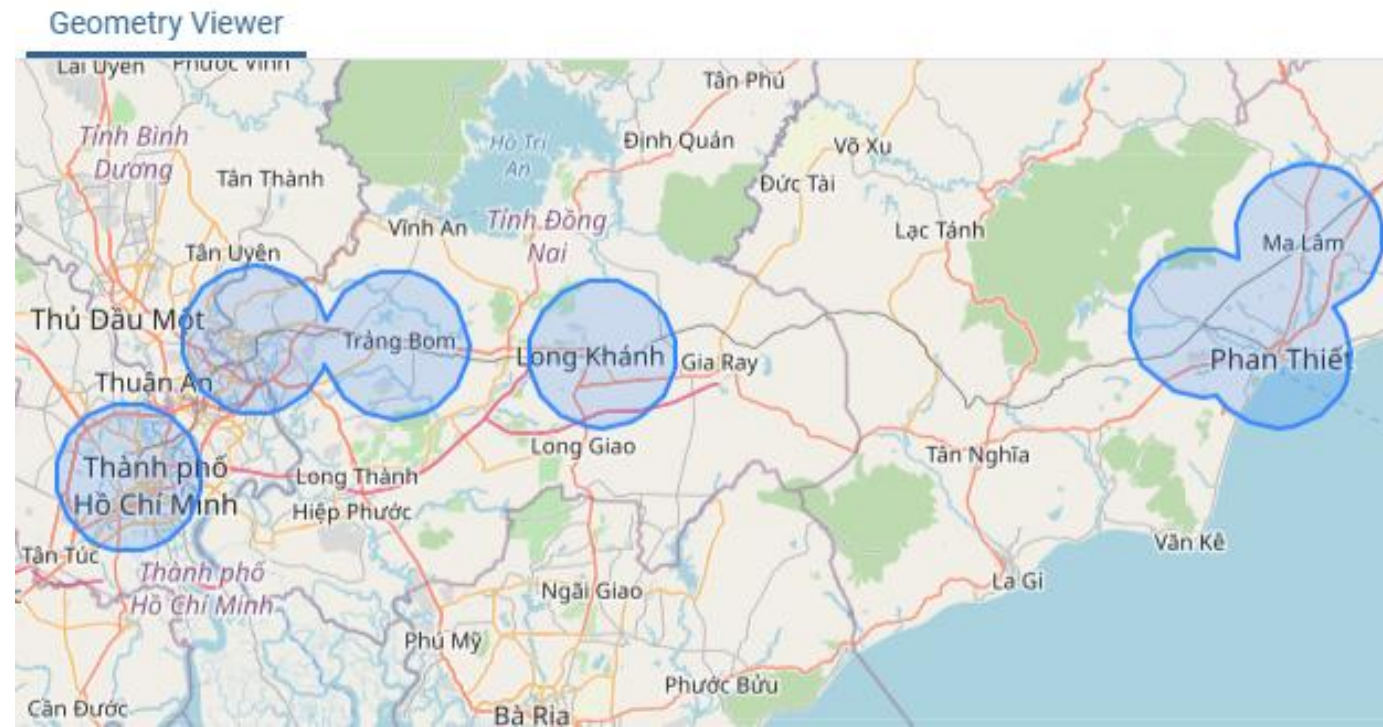


## SQL tạo dữ liệu không gian – Union

- ◆ **SELECT ST\_Union(ST\_Buffer(geom::Geography, 10000)::Geometry)**
- ◆ **FROM vietnam\_railway\_station\_point;**

Tạo lớp vùng đệm 10 km duy nhất cho lớp ga đường sắt (vietnam\_railway\_station\_point)?

	st_union	
	geometry	 
1	0106000020E6100...	

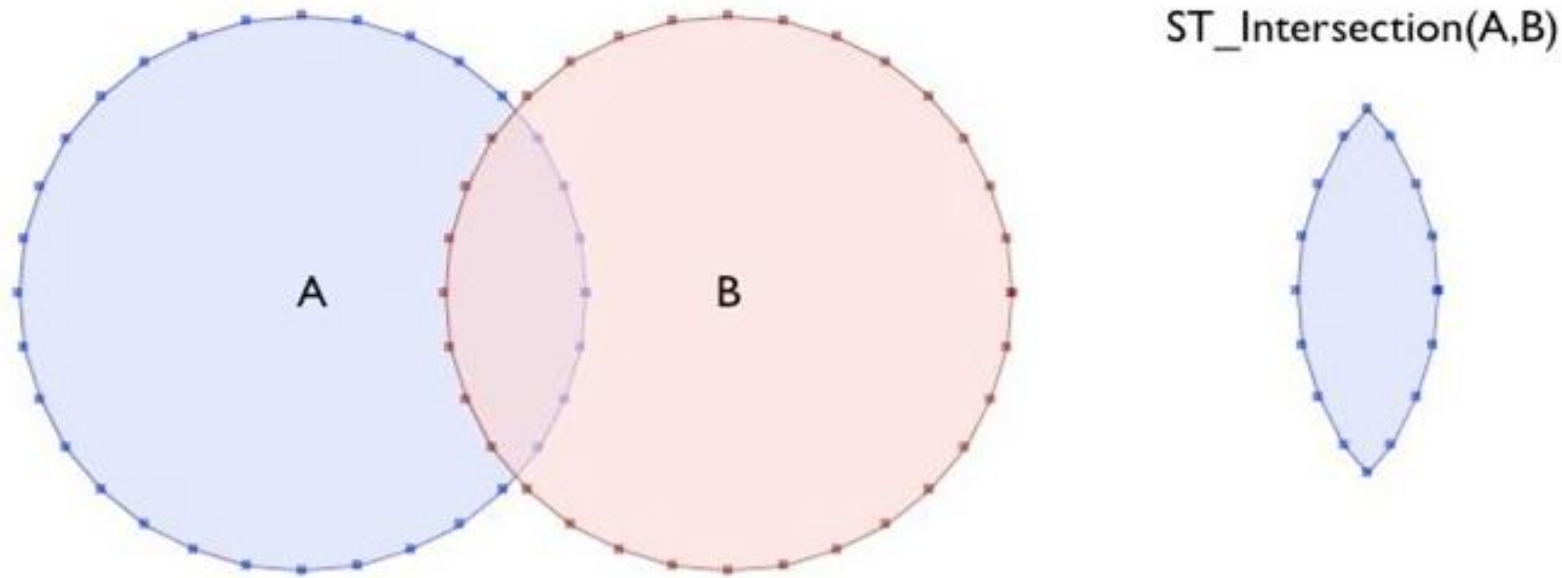




# SQL tạo dữ liệu không gian – Intersection

## ◆ ST\_Intersection(geometry A, geometry B)

◆ Trả về lớp vùng (hoặc đường, điểm) chung của hai lớp đầu vào.

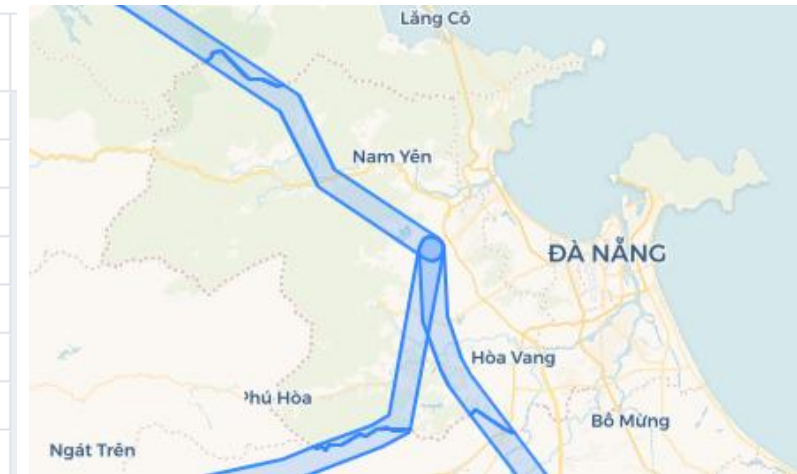


## SQL tạo dữ liệu không gian – Intersection

- ◆ **SELECT** p.Ten, **ST\_Intersection**(p.geom ::Geography, **ST\_Buffer**(e.geom ::Geography, 1000)) **AS** geom
- ◆ **FROM** vietnam\_province\_border\_polygon p,  
vietnam\_electricity\_transmission\_network\_line e
- ◆ **WHERE** e.Voltage = 500 **AND** **ST\_Intersects**(p.geom ::Geography, **ST\_Buffer**(e.geom ::Geography, 1000))
- ◆ **ORDER BY** p.Ten;

Tạo lớp vùng giao nhau giữa địa giới tỉnh (vietnam\_province\_border\_polygon) và vùng đệm 1 km của đường dây tải điện 500kV (vietnam\_electricity\_transmission\_network\_line) theo từng tỉnh thành?

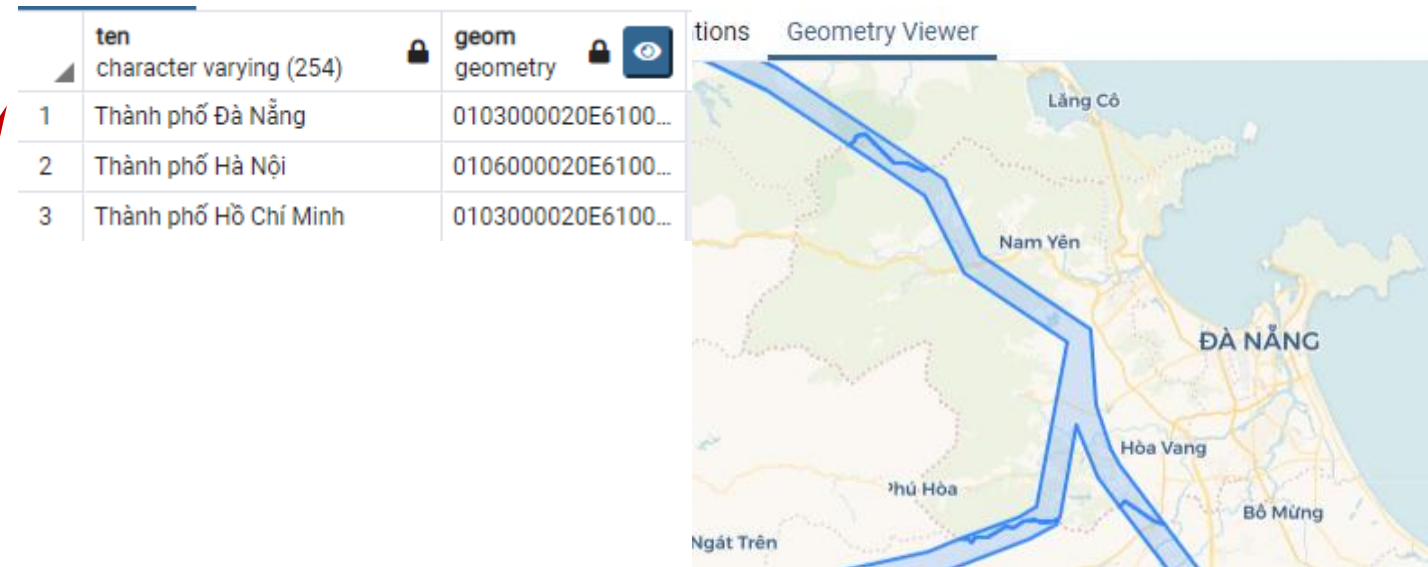
	ten	geom
	character varying (254)	geography
1	Thành phố Đà Nẵng	0103000020E61000...
2	Thành phố Đà Nẵng	0103000020E61000...
3	Thành phố Đà Nẵng	0103000020E61000...
4	Thành phố Hà Nội	0106000020E61000...
5	Thành phố Hà Nội	0103000020E61000...
6	Thành phố Hà Nội	0103000020E61000...
7	Thành phố Hà Nội	0103000020E61000...
8	Thành phố Hồ Chí Minh	0103000020E61000...



# SQL tạo dữ liệu không gian – Intersection (Union)

- ◆ **SELECT** p.Ten, **ST\_Union**(**ST\_Intersection**(p.geom ::Geography, **ST\_Buffer**(e.geom ::Geography, 1000))::Geometry) AS geom
- ◆ **FROM** vietnam\_province\_border\_polygon p,  
vietnam\_electricity\_transmission\_network\_line e
- ◆ **WHERE** e.Voltage = 500 AND **ST\_Intersects**(p.geom ::Geography, **ST\_Buffer**(e.geom ::Geography, 1000))
- ◆ **GROUP BY** p.Ten
- ◆ **ORDER BY** p.Ten;

Tạo lớp vùng giao nhau giữa địa giới tỉnh (vietnam\_province\_border\_polygon) và vùng đệm 1 km của đường dây tải điện 500kV (vietnam\_electricity\_transmission\_network\_line) theo từng tỉnh thành?

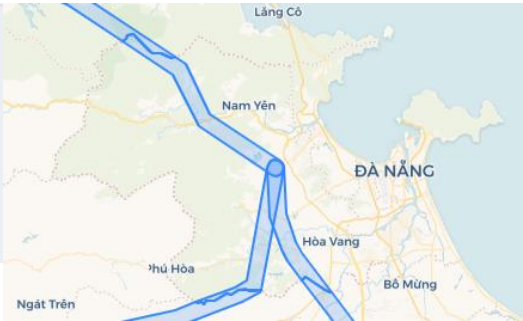


# SQL tạo dữ liệu không gian – Intersection

- ◆ **SELECT** p.Ten, **Sum(ST\_Area(ST\_Intersection(p.geom ::Geography, ST\_Buffer(e.geom ::Geography, 1000)))/10000)** AS DienTich\_ha
- ◆ **FROM** vietnam\_province\_border\_polygon p, vietnam\_electricity\_transmission\_network\_line e
- ◆ **WHERE** e.Voltage = 500 **AND ST\_Intersects**(p.geom ::Geography, ST\_Buffer(e.geom ::Geography, 1000))
- ◆ **GROUP BY** p.Ten
- ◆ **ORDER BY** p.Ten;

Tính diện tích giao nhau (ha) giữa địa giới tỉnh (vietnam\_province\_border\_polygon) và vùng đệm 1 km của đường dây tải điện 500kV (vietnam\_electricity\_transmission\_network\_line) theo từng tỉnh thành?

	ten character varying (254)	dientich_ha double precision
1	Thành phố Đà Nẵng	11451.0325126211
2	Thành phố Hà Nội	15000.0946777385
3	Thành phố Hồ Chí Minh	18441.8440463711
4	Tỉnh Bà Rịa - Vũng Tàu	547.225259996634



# SQL tạo dữ liệu không gian – Intersection (Union)

- ◆ **SELECT** p.Ten, **ST\_Area(ST\_Union(ST\_Intersection(p.geom ::Geography, ST\_Buffer(e.geom ::Geography, 1000))::Geometry)::Geography)/10000** AS **DienTich\_ha**
- ◆ **FROM** vietnam\_province\_border\_polygon p, vietnam\_electricity\_transmission\_network\_line e
- ◆ **WHERE** e.Voltage = 500 **AND** **ST\_Intersects**(p.geom ::Geography, ST\_Buffer(e.geom ::Geography, 1000))
- ◆ **GROUP BY** p.Ten
- ◆ **ORDER BY** p.Ten;

Tính diện tích giao nhau (ha) giữa địa giới tỉnh (vietnam\_province\_border\_polygon) và vùng đệm 1 km của đường dây tải điện 500kV (vietnam\_electricity\_transmission\_network\_line) theo từng tỉnh thành?

	ten	dientich_ha
	character varying (254)	double precision
1	Thành phố Đà Nẵng	10120.2680758218
2	Thành phố Hà Nội	14368.2844553792
3	Thành phố Hồ Chí Minh	16110.3002632368
4	Tỉnh Bà Rịa - Vũng Tàu	547.225259996634





## Bài tập 3: SQL tạo dữ liệu không gian

1. Tạo lớp mặt đường sắt dạng vùng với khổ rộng 1 m từ lớp đường sắt (vietnam\_railway\_line)?
2. Tính tổng diện tích (km<sup>2</sup>) mặt đường sắt dạng vùng với khổ rộng 1 m từ lớp đường sắt (vietnam\_railway\_line)?
3. Thống kê tổng chiều dài đường sắt (km) đi qua từng tỉnh thành (vietnam\_province\_border\_polygon), sắp xếp kết quả theo tổng chiều dài đường sắt giảm dần?

	tuyenduongsat character varying (254)	geom geography
1	Hà Nội - Hồ Chí Minh	0103000020E61000...
2	Hà Nội - Hải Phòng	0103000020E61000...
3	Hà Nội - Lạng Sơn	0103000020E61000...
4	Hà Nội - Lào Cai	0103000020E61000...
5	Hà Nội - Hồ Chí Minh	0103000020E61000...
6	Hà Nội - Hồ Chí Minh	0103000020E61000...
7	Hà Nội - Hồ Chí Minh	0103000020E61000...
8	Hà Nội - Hồ Chí Minh	0103000020E61000...
9	Cam Đường - Phố Lu	0103000020E61000...

	dientich double precision	geom geometry
1	2.61933004353524	0106000020E6100...

	ten character varying (254)	chieudaiduongsat double precision
1	Tỉnh Bình Thuận	174.959588126886
2	Tỉnh Quảng Bình	170.501751202759
3	Thành phố Hà Nội	149.351302154697
4	Tỉnh Khánh Hòa	148.644607655986
5	Tỉnh Bình Định	144.928171302591
6	Tỉnh Lạng Sơn	118.663812545158
7	Tỉnh Thừa Thiên Huế	107.734704256101
8	Tỉnh Lào Cai	105.36990127809
9	Tỉnh Thanh Hóa	102.660849845336



## SQL chỉ mục không gian

---

- ◆ Chỉ mục không gian giúp tăng tốc độ truy vấn không gian...
- ◆ Kết hai bảng với 10.000 dòng/bảng.

Không có chỉ mục	Có chỉ mục
$10.000 * 10.000 =$ <b>100.000.000</b> lượt so sánh	$10.000 + 10.000 =$ <b>20.000</b> lượt so sánh

## SQL chỉ mục không gian

---

◆ Để chứng minh điều đó... hãy xóa chỉ mục:

◆ **DROP INDEX** vietnam\_electricity\_transmission\_network\_line\_geom\_idx;

◆ Chạy truy vấn không gian:

◆ **SELECT** p.Ten, **Sum(ST\_Area (ST\_Intersection(p.geom ::Geography, ST\_Buffer(e.geom ::Geography, 1000)))/10000)** AS DienTich\_ha

◆ **FROM** vietnam\_province\_border\_polygon p,  
vietnam\_electricity\_transmission\_network\_line e

◆ **WHERE** e.Voltage = 500 **AND** ST\_Intersects(p.geom, e.geom)

◆ **GROUP BY** p.Ten;

~858 ms

## SQL chỉ mục không gian

---

### ◆ Tạo lại chỉ mục:

◆ **CREATE INDEX** vietnam\_electricity\_transmission\_network\_line\_geom\_idx  
**ON** vietnam\_electricity\_transmission\_network\_line **USING GIST** (geom);

### ◆ Chạy lại truy vấn không gian:

◆ **SELECT** p.Ten, **Sum**(**ST\_Area** (**ST\_Intersection**(p.geom ::Geography,  
**ST\_Buffer**(e.geom ::Geography, 1000)))/10000) **AS** DienTich\_ha

◆ **FROM** vietnam\_province\_border\_polygon p,  
vietnam\_electricity\_transmission\_network\_line e

◆ **WHERE** e.Voltage = 500 **AND** **ST\_Intersects**(p.geom, e.geom)

◆ **GROUP BY** p.Ten;

~470 ms

## SQL chỉ mục không gian

---

### ◆ Các hàm không gian sử dụng chỉ mục:

- ◆ ST\_Intersects()
- ◆ ST\_Contains()
- ◆ ST\_Within()
- ◆ ST\_DWithin()
- ◆ ST\_ContainsProperly()
- ◆ ST\_CoveredBy()
- ◆ ST\_Covers()
- ◆ ST\_Overlaps()
- ◆ ST\_Crosses()
- ◆ ST\_DFullyWithin()

### ◆ ST\_3DIntersects()

### ◆ ST\_3DDWithin()

### ◆ ST\_3DDFullyWithin()

### ◆ ST\_LineCrossingDirection()

### ◆ ST\_OrderingEquals()

### ◆ ST\_Equals()



## Bài tập kiểm tra

---

- ◆ Điền câu trả lời cho các câu hỏi trong Google Form  
<https://forms.gle/XK8nxFp3huB9jA1S8?>