

Otro tipo de uniones Espaciales en PostGIS

Cristhian Camilo Cante Huertas

20201025024

Bases de datos espaciales – 2025 1

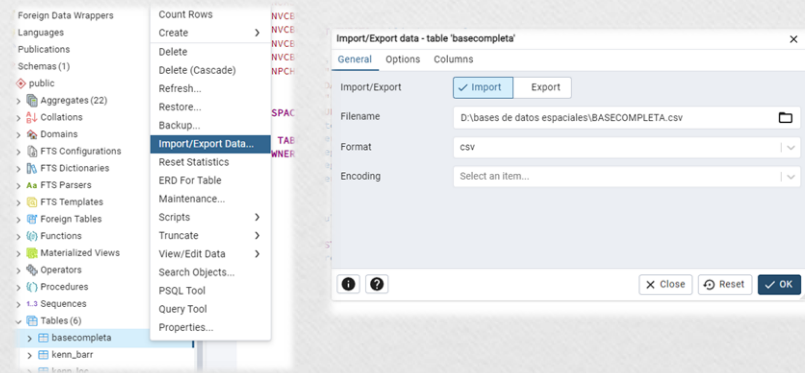
Características de las uniones espaciales

- Permiten combinar datos espaciales con atributos
- Basadas en relaciones espaciales como intersección o contención
- Útiles para análisis geográficos complejos

Creación de tabla de datos de encuesta sin espacializar

- Se crea la tabla, definiendo su nombre, el tipo de datos y el nombre de sus columnas.
- Se definen los parámetros de importación (encabezado, not null, etc..)
- Se busca el archivo a importar

```
-- DROP TABLE IF EXISTS public.basecompleta;  
  
CREATE TABLE IF NOT EXISTS public.basecompleta  
(  
    "MPI0" integer,  
    "COD_LOCALIDAD" integer,  
    "NOMBRE_LOCALIDAD" character varying(255) COLLATE pg_catalog."default",  
    "COD_UPZ_GRUPO" integer,  
    "NOMBRE_UPZ_GRUPO" character varying(255) COLLATE pg_catalog."default",  
    "NVCBP11AA" integer,  
    "NVCBP5" integer,  
    "NVCBP14F" integer,  
    "NVCBP15C" integer,  
    "NPCHP4" integer  
)  
  
TABLESPACE pg_default;  
  
ALTER TABLE IF EXISTS public.basecompleta  
OWNER to postgres;
```



Se adaptan los datos de la base de datos para su posterior unión

- Se cambian los nombres de las columnas por nombres más dicientes y se eliminan los campos nulos
- Se transforman algunas variables para su análisis
- Se crea una tabla con datos agregados a nivel UPZ para poder realizar la unión

```
CREATE TEMP TABLE bogota_temp AS
SELECT
  "RP10",
  "COD_LOCALIDAD" AS "Cod_Localidad",
  "NOMBRE_LOCALIDAD" AS "Nombre_Loc",
  "COD_UPZ_GRUPO" AS "Cod_UPZ",
  "NOMBRE_UPZ_GRUPO" AS "Nombre_UPZ",
  "NVCBP11AA" AS "Estrato",
  "NVCBP5" AS "IluNoct",
  "NVCBP14F" AS "Drogas",
  "NVCBP15C" AS "Inseguridad",
  "NVCBP4" AS "NivEdu"
FROM
  basecompleta
WHERE
  "RP10" = 11001;

-- Paso 2: Eliminar filas con valores nulos (complete.cases)
DELETE FROM Bogota_temp
WHERE
  "Estrato" IS NULL OR
  "IluNoct" IS NULL OR
  "Drogas" IS NULL OR
  "Inseguridad" IS NULL OR
  "NivEdu" IS NULL;

CREATE TEMP TABLE Bogota_procesada AS
SELECT
  *,
  CASE WHEN "Drogas" = '1' THEN 1 ELSE 0 END AS "SiDrogas",
  CASE WHEN "Inseguridad" = '1' THEN 1 ELSE 0 END AS "SiInseguridad",
  CASE WHEN CAST("Estrato" AS INTEGER) >= 4 THEN 1 ELSE 0 END AS "EstratoAlto",
  CASE WHEN CAST("NivEdu" AS INTEGER) >= 9 THEN 1 ELSE 0 END AS "EduSup"
FROM
  Bogota_temp;

-- Paso 4: Agrupar y resumir para crear los índices (group_by y summarise)
CREATE TABLE resumen_upz AS -- Creamos una tabla persistente para los resultados
SELECT
  "Cod_UPZ",
  "Cod_Localidad",
  "Nombre_Loc",
  "Nombre_UPZ",
  AVG("SiDrogas") * 100 AS porcentaje_drogas,
  AVG("SiInseguridad") * 100 AS porcentaje_inseguridad,
  AVG("EstratoAlto") * 100 AS porcentaje_estratoAlto,
  AVG("EduSup") * 100 AS porcentaje_eduSup
FROM
  Bogota_procesada
GROUP BY
  "Cod_UPZ",
  "Cod_Localidad",
  "Nombre_Loc",
  "Nombre_UPZ";

ALTER TABLE upz_kennedy
ALTER COLUMN "CODIGO_UPZ" TYPE INTEGER
USING "CODIGO_UPZ"::integer;
```


Se crea la tabla con datos y geometrías

- Se utiliza un JOIN para crear una tabla con los datos agregados del resumen por UPZ y las geometrías del mapa de UPZ, esto por medio de sus códigos
- Se hace el contraste entre las tablas iniciales y finales

```
CREATE TABLE upz_kennedy_con_indices AS
SELECT
  u.*, -- Selecciona todas las columnas de la tabla upz_kennedy
  r.porcentaje_drogas,
  r.porcentaje_inseguridad,
  r.porcentaje_EstratoAlto,
  r.porcentaje_EduSup,
  r."Cod_Localidad" AS cod_loc,
  r."Nombre_Loc" AS Nom_loc
FROM
  upz_kennedy u |
LEFT JOIN
  resumen_upz r ON u."CODIGO_UPZ" = r."Cod_UPZ"; -- La condición de unión
```

Tables (8)

- > basecompleta
- > kenn_barr
- > kenn_loc
- > kenn_vias
- > spatial_ref_sys
- > upz_kennedy

Tables (8)

- > basecompleta
- > kenn_barr
- > kenn_loc
- > kenn_vias
- > resumen_upz
- > spatial_ref_sys
- > upz_kennedy
- > upz_kennedy_con_indices

Se consulta por los barrios con mayor nivel de educación

- Se seleccionan las columnas que se desean obtener: El nombre del barrio, la UPZ y el porcentaje de educación.
- Se pone la condición de intersección entre la UPZ y los barrios
- Se añade la condición para solo seleccionar el registro de mayor valor

Query Query History

```
1 SELECT
2   b.barrioconu AS nombre_barrio,
3   u."NOMBRE" AS nombre_upz,
4   u.porcentaje_edusup AS porcentaje_educacion_superior
5 FROM
6   upz_kennedy_con_indices u
7 JOIN
8   kenn_barr b ON ST_Intersects(b.geom, u.geom)
9 WHERE
10  u."NOMBRE" = (
11    SELECT
12      "NOMBRE"
13    FROM
14      upz_kennedy_con_indices
15    WHERE
16      porcentaje_edusup IS NOT NULL
17    ORDER BY
18      porcentaje_edusup DESC
19    LIMIT 1
20  );
```

Data Output Messages Notifications

	nombre_barrio character varying (254)	nombre_upz character varying (250)	porcentaje_educacion_superior numeric
1	Bavaria I Sector	AMERICAS	32.38770685579196217500
2	Glorieta de las Americas	AMERICAS	32.38770685579196217500
3	Santa Rosa del Carvajal	AMERICAS	32.38770685579196217500
4	Milenta II Sector	AMERICAS	32.38770685579196217500
5	El Triángulo	AMERICAS	32.38770685579196217500
6	Centro Americas	AMERICAS	32.38770685579196217500
7	El Triunfo de las Americas	AMERICAS	32.38770685579196217500
8	Hipotecho Occidental II	AMERICAS	32.38770685579196217500
9	La Igualdad	AMERICAS	32.38770685579196217500
10	Las Americas Occidental I	AMERICAS	32.38770685579196217500
11	Alferez Real	AMERICAS	32.38770685579196217500
12	Las Americas La Triguera	AMERICAS	32.38770685579196217500
13	Floresta del Sur	AMERICAS	32.38770685579196217500

Total rows: 48 Query complete 00:00:00.050

Otra manera de hacerlo

- En lugar de usar una relación de intersección, ahora se usa una de contenencia, donde la UPZ debe contener a los barrios para que estos aparezcan en la consulta
- Ahora solo 35 registros coincidieron con la consulta, en lugar de 48.

Query Query History

```
1 SELECT
2   b.barriocomu AS nombre_barrio,
3   u."NOMBRE" AS nombre_upz,
4   u.porcentaje_edusup AS porcentaje_educacion_superior
5 FROM
6   upz_kennedy_con_indices u
7 JOIN
8   kenn_barr b ON ST_Contains(u.geom, ST_Centroid(b.geom))
9 WHERE
10  u."NOMBRE" = (
11    SELECT
12      "NOMBRE"
13    FROM
14      upz_kennedy_con_indices
15    WHERE
16      porcentaje_edusup IS NOT NULL
17    ORDER BY
18      porcentaje_edusup DESC
19    LIMIT 1
20  );
```

Data Output Messages Notifications

	nombre_barrio character varying (254)	nombre_upz character varying (250)	porcentaje_educacion_superior numeric
1	Glorieta de las Americas	AMERICAS	32.38770685579196217500
2	Santa Rosa del Carvajal	AMERICAS	32.38770685579196217500
3	Milenta II Sector	AMERICAS	32.38770685579196217500
4	Centro Americas	AMERICAS	32.38770685579196217500
5	El Triunfo de las Americas	AMERICAS	32.38770685579196217500
6	Hipotecho Occidental II	AMERICAS	32.38770685579196217500
7	La Igualdad	AMERICAS	32.38770685579196217500
8	Las Americas Occidental I	AMERICAS	32.38770685579196217500
9	Alferez Real	AMERICAS	32.38770685579196217500
10	Las Americas La Triguera	AMERICAS	32.38770685579196217500
11	Floresta del Sur	AMERICAS	32.38770685579196217500
12	Nueva Marsella II Sector	AMERICAS	32.38770685579196217500
13	Plá X - Urb. El Carmen	AMERICAS	32.38770685579196217500

Se añade otra tabla con el fin de realizar otro tipo de consulta

- Se añade la tabla de estaciones Transmilenio

- Se pone a prueba la hipótesis de 2 maneras distintas

```
1 SELECT
2   u."NOMBRE" AS nombre_upz,
3   COUNT(e.id) AS numero_estaciones,
4   u.porcentaje_inseguridad
5 FROM
6   upz_kennedy_con_indices u
7 JOIN
8   kenn_estransr e ON ST_Contains(u.geom, e.geom)
9 WHERE
10  u.porcentaje_inseguridad IS NOT NULL
11 GROUP BY
12  u."NOMBRE", u.porcentaje_inseguridad
13 ORDER BY
14  numero_estaciones DESC
15 LIMIT 1;
```

Data Output Messages Notifications

	nombre_upz character varying (250)	numero_estaciones bigint	porcentaje_inseguridad numeric
1	AMERICAS	2	64.83451536643026004700

> kenn_estransr

```
1 SELECT
2   u."NOMBRE" AS nombre_upz,
3   COUNT(e.id) AS numero_estaciones,
4   u.porcentaje_inseguridad
5 FROM
6   upz_kennedy_con_indices u
7 LEFT JOIN
8   kenn_estransr e ON ST_Contains(u.geom, e.geom)
9 WHERE
10  u."NOMBRE" = (
11    SELECT
12      "NOMBRE"
13    FROM
14      upz_kennedy_con_indices
15    WHERE
16      porcentaje_inseguridad IS NOT NULL
17    ORDER BY
18      porcentaje_inseguridad DESC
19    LIMIT 1
20  )
21 AND u.porcentaje_inseguridad IS NOT NULL
22 GROUP BY
23  u."NOMBRE", u.porcentaje_inseguridad;
```

Data Output Messages Notifications

	nombre_upz character varying (250)	numero_estaciones bigint	porcentaje_inseguridad numeric
1	CORABASTOS	1	91.60387513455328310000

Preguntas

- ¿Por qué se necesitaron 2 tablas para crear la tabla “upz_kennedy_con_índices”?
- ¿Qué se tomó de cada una?
- ¿Por qué la segunda consulta se realizó de 2 maneras?