Tema 4

# Práctica: Importación de datos espaciales y uso de comandos SQL para el análisis de una base de datos espacial

## **Importar ficheros shapefile**

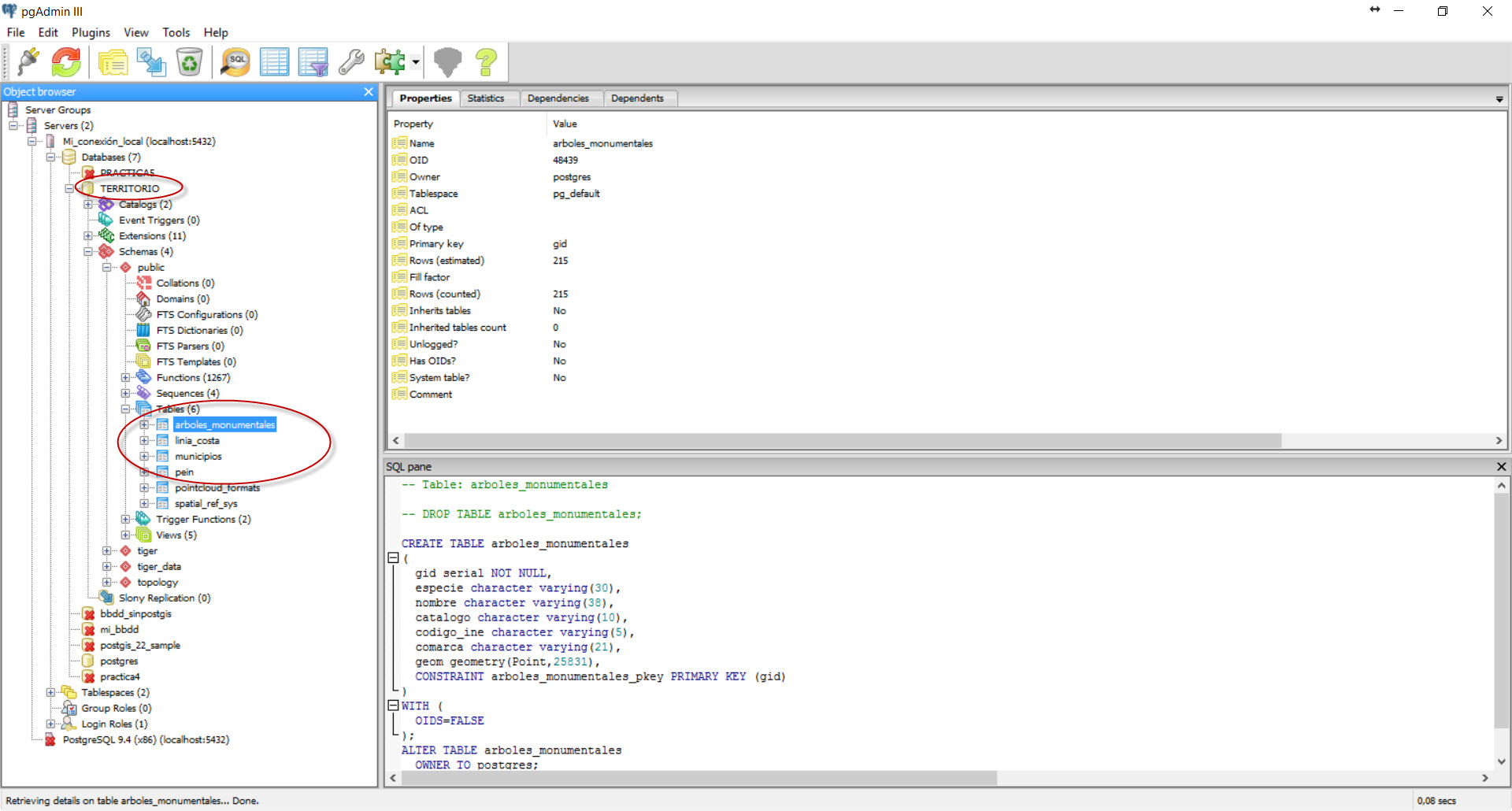
shp2pgsql -W "latin1" "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\datos\_practicafinal\arboles\_monumentales.shp" ARBOLES\_MONUMENTALES > "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\Resultats\arboles\_monumentales.sql"

shp2pgsql -W "latin1" "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\datos\_practicafinal\linea\_costa.shp" LINIA\_COSTA > "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\Resultats\linea\_costa.sql"

shp2pgsql -W "latin1" "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\datos\_practicafinal\municipios.shp" MUNICIPIOS > "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\Resultats\municipios.sql"

shp2pgsql -W "latin1" "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\datos\_practicafinal\pein.shp" PEIN > "C:\IP\_DIARI\CURS\_POSTGIS\Tema\_4\Resultats\pein.sql"

## **Captura de pantalla importación**



## **Actualitzar Sistema de referència a ETRS89 (25831)**

SELECT UpdateGeometrySRID('arboles\_monumentales','geom',25831);

SELECT UpdateGeometrySRID('linia\_costa','geom',25831);

SELECT UpdateGeometrySRID('municipios','geom',25831);

SELECT UpdateGeometrySRID('pein','geom',25831);

## **Respuestas comandas SQL**

**1) Obtener un listado donde aparezca la distancia media entre todos los árboles monumentales de la misma especie.**

SELECT t1.especie, avg(ST\_Distance(t1.geom, t2.geom)) as Media FROM arboles\_monumentales t1, arboles\_monumentales t2 GROUP BY t1.especie ORDER BY t1.especie;

**2) Obtener un listado con los municipios que tienen parte de su territorio dentro de algún espacio PEIN de más de 20.000ha. En el listado debe aparecer, como mínimo, el nombre del municipio, el nombre del espacio pein y la superficie del espacio PEIN.**

SELECT municipios.nombre as NomMunicipi, AreaGran.nombre as NomPein, St\_Area(AreaGran.geom)/1000 as AreaPein

FROM municipios,

(

SELECT nombre, geom

FROM pein

WHERE (St\_Area(pein.geom)/10000 > 20000)

)as AreaGran

WHERE St\_Intersects(municipios.geom, AreaGran.geom)=true

ORDER BY NomMunicipi;

**3) Obtener un listado donde aparezca una relación de municipios con el número de árboles monumentales que le pertenecen.**

SELECT ContarArbres.nombre as Municipi, count(\*) as NumArbres

from

(select municipios.nombre from municipios, arboles\_monumentales

Where st\_contains(municipios.geom, arboles\_monumentales.geom)=true

) as ContarArbres

GROUP BY ContarArbres.nombre

ORDER BY ContarArbres.nombre;

**4) Obtener un listado de las comarcas de Catalunya junto con el número de árboles monumentales que contiene de cada especie.**

SELECT t1.comarca, t1.especie, count(\*)

FROM arboles\_monumentales t1

GROUP BY t1.comarca, t1.especie

ORDER BY t1.comarca;

**5) Elaborar un listado de los árboles monumentales que se encuentran a más de 2km de cualquier espacio de interés natural.**

SELECT arboles\_monumentales.gid, arboles\_monumentales.nombre

FROM arboles\_monumentales,

(

SELECT St\_Union(pein.geom) as geom

FROM pein

) As TotPein

WHERE St\_Intersects(TotPein.geom, St\_Buffer(arboles\_monumentales.geom, 2000))=false

ORDER BY arboles\_monumentales.nombre;

**6) Establecer una zona de influencia marítima a 1.000 metros de la línea de costa en dirección interior. Es decir, la zona de influencia no debe adentrase en el mar.**

SELECT St\_Intersection(UnionMun.geom, Buffer.geom) as Influencia

FROM

(

SELECT St\_Union(municipios.geom) As geom

FROM municipios

) As UnionMun,

(

SELECT St\_Buffer(linia\_costa.geom,1000) As geom

FROM linia\_costa

) As Buffer

;

**7) Obtener un listado donde aparezca la densidad de árboles monumentales por unidad de superficie para los municipios que tienen algún árbol monumental.**

SELECT municipios.nombre as NomMunicipi, ResultatContar.NumArbres, ResultatContar.NumArbres/St\_Area(municipios.geom) as Densitat

FROM

municipios,

(

SELECT ContarArbres.municipio, count(\*) as NumArbres

from

(select municipios.municipio, municipios.geom from municipios, arboles\_monumentales

Where st\_contains(municipios.geom, arboles\_monumentales.geom)=true

) as ContarArbres

GROUP BY ContarArbres.municipio

ORDER BY ContarArbres.municipio

) As ResultatContar

WHERE municipios.municipio = ResultatContar.municipio

ORDER BY municipios.nombre;

**8) Obtener un listado donde aparezcan las provincias de Catalunya con los km's de costa de cada una de ellas.**

SELECT Provincies.provincia as CodiProvincia, St\_Length2D(St\_Intersection(Provincies.geom,linia\_costa.geom))/1000 as KmCosta

FROM linia\_costa,

(

SELECT provincia, St\_Collect(municipios.geom) as geom

FROM municipios

GROUP BY provincia

) as Provincies

ORDER BY Provincies.provincia;

**9) Realizar un buffer de 500 metros sobre los islotes de la costa catalana. Los islotes se pueden obtener a partir de las geometrías de los municipios. Ningún islote estará a más de 5000 de la línea de costa.**

SELECT St\_Buffer(DumpGeom.geom,500)

FROM linia\_costa,

(

SELECT (St\_Dump(St\_Union(municipios.geom))).geom

FROM municipios

) AS DumpGeom

WHERE St\_Distance(St\_Centroid(DumpGeom.geom),linia\_costa.geom)<5000;

**10) Obtener el municipio con más superficie de PEIN en su territorio en proporción a su superficie. Muestra el resultado con %.**

SELECT municipios.nombre, (((SUM(St\_Area(municipios.geom))-SUM(St\_Area(St\_Difference(municipios.geom, pein.geom))))\*100)/SUM(St\_Area(municipios.geom))) AS TanPerCent

FROM municipios,pein

WHERE St\_Intersects(municipios.geom, pein.geom)

GROUP BY municipios.nombre

ORDER BY TanPerCent DESC;