21. Adatbázis-kezelés

A megoldáshoz az SQL nyelv ismerete szükséges.

Nyissuk meg a HUNGARY. SQLITE állományt, hajtsuk végre az alábbi lekérdezéseket!

Megoldás

A fájlt most ne vektoros rétegként nyissuk meg, hanem adatbázisként: Database menü DBManager.

A bal oldali boxban a SpatiaLite/Geopackage-re jobb egérgombbal kattintunk és létrehozunk egy New Connection-t a HUNGARY.SQLITE-tal.

Magyarázat Mi az SQLite? https://www.sqlite.org/



Az SQLite önálló, kisméretű C forrású programkönyvtárként megvalósított relációs adatbáziskezelő-rendszer. Előnye, hogy nem kliens-szerver architektúrájú, hanem a teljes adatbázis (definíciók, táblák, indexek és adatok is) egyetlen platformfüggetlen fájlban tárolódnak a programot futtató számítógépen. A térbeli kiterjesztése a SpatiaLite, ezt használja a QGIS.

Nézzük meg az adatbázis metaadatait! Ehhez nyissuk ki a köteg jelét. Információt kaphatunk a táblák szerkezetéről az Info fülön. Milyen mezők találhatók benne, milyen a bennük tárolt adatok típusa, van-e beállított index. A Table-nél láthatjuk a táblázatot az adatokkal, a Preview-nál pedig az adatok előnézetét, ha van. Ha lekérdezést hajtunk végre a táblát az ablak ikonsorában található Refresh gombbal frissíthetjük, lekérdezéshez nyissuk meg a következő ikonnal a lekérdező ablakot (SQL window).

A lekérdezést a felső boxba gépelhetjük, és az Execute (F5) gombbal hajtható végre. Alul látható az eredmény.

1) Feladat

A geobox tábla a geom_wgs84 mezőjében földrajzi koordináták találhatók. Váltsuk át ezeket EOVbe! Hozzunk létre egy új mezőt, készítsünk hozzá térbeli indexet, és töltsük fel a létrehozott új mezőt adatokkal!

1) Megoldás

SELECT AddGeometryColumn('geobox', 'geom_eov', 23700, 'POINT', 3); Új oszlop hozzáadása. A függvény paraméterezése: (táblanév, oszlop neve, a vetület EPSG száma, a geometria típusa, háromdimenziós geometria)

majd készítsük indexet. SELECT CreateSpatialIndex('geobox', 'geom_eov'); A függvény paraméterezése: (tábla neve, az indexet viselő oszlop neve) végül töltsük fel adattal:

UPDATE geobox SET geom_eov=transform(geom_wgs84, 23700); A transform paraméterezése: (mik a bemenő koordináták, és mibe transzformálunk (EPSG szám))

Mi az az EPSG szám?

Egy egységes nevezékrendszer a térbeli koordinátarendszerek azonosítására. A legfontosabbak: WGS84: EPSG 4326, HD72/EOV: EPSG23700, WebMercator: EPSG 3857.

http://spatialreference.org/

A vetületek esetén a SpatiaLite a proj4 függvénykönyvtárral dolgozik, ebben megtalálható az EOV, amelynek EPSG száma ismert.

2) Feladat

Hozzunk létre geometriát (megyék poligonjai) a county táblába! Használjuk ehhez a muni táblát is! A vetület EOV.

Vagyis hozzunk létre a poligonnak egy új oszlopot a county táblában, készítsünk hozzá térbeli indexet, és végül töltsük fel adattal!

2) Megoldás

SELECT AddGeometryColumn('county', 'geom', 23700, 'POLYGON', 2); Új oszlop hozzáadása. A függvény paraméterezése: (táblanév, oszlop neve, a vetület EPSG száma, a geometria típusa, kétdimenziós geometria)

majd térbeli indexelés: SELECT CreateSpatialIndex('county', 'geom'); A függvény paraméterezése: (tábla neve, az indexet viselő oszlop neve)

Az oszlop benépesítése:

UPDATE county SET geom=(SELECT ST_Union(geom) FROM muni WHERE muni.county_id=county.county_id);

Allekérdezéssel hozzuk létre a geometriát. Az allekérdezésben a select után kiíratjuk a megye geometriáját. Ehhez előbb össze kell kapcsolnunk két mezőt: a megye azonosítóját a település táblából (muni.county_id), a megye azonosítójával a megye táblából (county.county_id). A települések geometriáját pedig ez alapján egyesítjük. Ehhez a ST_UNION(geometriát tartalmazó oszlop neve) függvényt használjuk.

3) Feladat

Írassa ki, mi a geoláda neve és melyik településen található! Írassuk ki a külföldi geoládákat is! Írassa ki a geoládát nem tartalmazó településeket is! Számoljuk meg hány geoláda van egy településen, rendezzünk a település neve szerint! Írassa ki azokat a településeket, ahol nincs geoláda!

3) Megoldás

SELECT geobox.name, muni.name FROM geobox, muni WHERE ST_Contains(muni.geom, geobox.geom_eov);

Az ST_Contains() függvény megvizsgálja, hogy az adott poligon tartalmazza-e a megadott pontot.

másként, join-nal:

SELECT geobox.name, muni.name FROM geobox JOIN muni ON ST_Contains(muni.geom, geobox.geom_eov);

Külföldi geoládák bevonása a lekérdezésbe

SELECT geobox.name, muni.name FROM geobox LEFT JOIN muni ON ST_Contains(muni.geom, geobox.geom_eov);

A geoládát nem tartalmazó települések bevonása a lekérdezésbe:

SELECT geobox.name, muni.name FROM muni LEFT JOIN geobox ON ST_Contains(muni.geom, geobox.geom_eov);

Geoládák darabszáma településenként, települések neve szerint rendezve: SELECT muni.name, count(*) AS gldb FROM muni JOIN geobox ON ST_Contains(muni.geom, geobox.geom_eov) GROUP BY muni.name ORDER BY 1;

Geoládák darabszáma településenként, azok is, ahol egy sincs:

SELECT muni.name, count(geobox.name) AS gldb FROM muni LEFT JOIN geobox ON ST_Contains(muni.geom, geobox.geom_eov) GROUP BY muni.name ORDER BY 1;

4) Feladat

Válogassuk le a következő települések 10 km-es pufferzónáját! Székesfehérvár, Szeged, Pécs, Debrecen, Győr, Miskolc, Veszprém, Szombathely, Györ, Kecskemét, Salgótarján! Hozzunk belőle létre egy új térképi nézetet!

4) Megoldás

select id, name, ST_Buffer(geom, 10000) as buffgeom from muni where name in ('Székesfehérvár', 'Szeged', 'Pécs', 'Debrecen', 'Győr', 'Miskolc', 'Veszprém', 'Szombathely', 'G yör', 'Kecskemét', 'Salgótarján');

Az ST_Buffer() függvénnyel létrehozható pufferzóna egy térképi elemhez, pl. poligonhoz. Első paraméter a geometria, amelyhez a pufferzóna készül, a második a pufferzóna szélessége méterben. IN(elem1,elem2,elem3...) Csak a megadott elemek leválogatása a táblából.

A térkép nézet létrehozása: Kattintsunk a Create view gombra, Load as new layer, és a geometry column-nál válasszuk a buffgeom oszlopot. Végül a Load now! gombbal a QGIS canvasre betölti a geometriát.

5) Feladat

Írassuk ki a Székesfehérvárral szomszédos települések nevét!

5) Megoldás

select name from muni where ST_Touches(geom, (select geom from muni where name='Székesfehérvár'));

Az ST_Touches függvény első és második paramétere az összehasonlítandó geometriák. Visszatérési értéke true v. false. Vagyis érintik-e egymást vagy sem. Székesfehérvár geometriáját egy allekérdezéssel tudjuk kinyerni az összehasonlításhoz.

6) Feladat

Nézzük meg, melyik ládák esnek Fehérvártól egy napi járásra (30 km-en belül)!

6) Megoldás

Select name from geobox where st_intersects (geom_eov,(select st_buffer(geom, 30000) from muni where name="Székesfehérvár"));

Generáljunk egy 30 000 méter széles pufferzónát a Székesfehérvár poligon köré az ST_Buffer(geometria, pufferzóna szélessége). Nézzük meg, hogy ennek a pufferzónának a kerületét melyik ládák keresztezik (ST_Intersects(láda, pufferzóna)). Ha az átfedés igaz, akkor kiírja a láda nevét.



Az adatért és a feladatokért ezúton fejezem ki hálás köszönetemet Nagy Gábornak!