Önálló laboratórium beszámoló

Dolgozat címe: .....

.....

Konzulens(ek) neve: .....

(Külső cég neve: .....

címe:.....

A Hallgató a kitűzött feladatot megfelelő színvonalon és a kiírásnak megfelelően

teljesítette nem teljesítette

_____

Konzulens aláírása

Hallgató neve: .....

Képzés: .....

Leadás dátuma: .....

Pázmány Péter Katolikus Egyetem Információs Technológiai Kar

GSM alapú helymeghatározás

Nagy mennyiség adat tárolása, feldolgozása valamint megjelenítése GSM alapú helymeghatározáshoz

Készítette: Réti Dániel

Konzulens: Tihanyi Attila

PPKE-ITK
2012.

Tartalomjegyzék

1 Bevezetés

Az önálló laboratórium projektemen egy népes csapatban dolgoztam, ezért tevékenységemet nem kellett az egészen alapoktól elkezdeni. Rengeteg adat, információ állt a rendelkezésemre amikor tavasszal csatlakoztam a csapathoz, melynek munkáját Tihanyi Attila vezette, felügyelte, segítette.

A csapat diáksága több felsőbb éves hallgatóból állt, akik már régóta foglalkoztak a témával, és 6 BSC-s hallgató lett újonnan tagja, köztük én is. Mivel nekünk újoncoknak az alapoktól kellett kezdenünk, a kutatási munkánk eleje az eddig elért eredmények, elvégzett mérések és a technológiai kihívások megismerésével kezdődött.

A cél egy működő helymeghatározó rendszer kifejlesztése, amely épületen belül és kívül a piacon is (jelenleg, vagy a közeljövőben) kapható mobiltelefonok használatával megfelelő pontosságú adatot biztosít a pozíciónkról. Ezzel elkerülhető a drága GPS készülékek megvásárlása, a mobiltelefon manapság majdnem minden ember zsebében megtalálható, illetve erre a szolgáltatásra további értékes szolgáltatások építhetők. Felhasználható például a családtagok, barátok helyzetét figyelő, folyamatosan frissítő rendszerként, amely a mobiltelefonokat használja a pozíciók lekérdezésére, egymás közti adatátvitelre, városba érkező(k) kalauzolására, látogatási tervek készítésére, világtalan emberek segítségére.

A technológiai megoldások kutatása során felmerült, hogy a helymeghatározáshoz jól működő, földrajzi koordinátákat jól kezelő adatbázis szükséges és erre a célra a legjobb egy földrajzi adatbázis-kezelő rendszer lehet. Egy ilyen adatbázis gyorsan képes nagy mennyiségű adatokat kezelni és feldolgozni a saját földrajzi célokra fejlesztett beépített függvényei segítségével, amellyel egy hagyományos adatbázis-kezelő rendszer nem rendelkezik.

Az én feladatom klasszifikációs eljárások vizsgálatára tesztelési eljárás írása volt, illetve az elkészített eljárás alapján algoritmusok tesztelése az adatbázis méretének csökkentése céljából.

2 Háttér

Ide kérek egy roved összefoglalót!

2.1 Mobil helymeghatározás

A csapat egyik korábbi publikációból vett bevezető szöveg jól leírja a feladatot: „A technika fejlődésével, és a lehetséges szolgáltatások bővülésével egyre nő az igény arra, hogy minél pontosabban, gyorsabban, és olcsóbban meg tudjuk határozni saját, vagy esetleg társunk tartózkodási helyét. Erre a műholdas rendszerek mellett a GSM hálózat is alkalmat kínál. A GSM hálózat működési tulajdonságainak kihasználásával szolgáltatás építhető, amely alkalmas helymeghatározásra. A felépített rendszer pontossága városi környezetben a GPS-hez mérhető. Ezzel alkalom nyílik személyi, gyalogos navigációra, akár beltéri környezetben is, ahol a műholdas helymeghatározás nem működik.
A rendszer helymeghatározásra alkalmas GSM hálózat felhasználásával működik, de a szokásos rendszerek hátrányainak kiküszöbölését is megvalósítja. A GPS hálózat üzemeltetési bizonytalansága, és a GPS készülékek magas ára nem okoz gondot, hiszen a kialakított rendszer GPS-hez hasonló pontosságú adatokat GPS felhasználása nélkül biztosítja. GSM szolgáltató közreműködése nélkül igénybe vehető, így biztosan nem tartozik hozzá számla és fizetési kötelezettség. A GSM szolgáltató nem is vesz részt a folyamatban, hiszen a szolgáltatás felépítése ezt nem teszi szükségessé. A helymeghatározás során a GSM hálózat pillanatnyi sugárzási karakterisztikáját használjuk fel, és az adatok szolgáltató által történő kismértékű megváltoztatása csak némi pontosság romlást eredményez, de nem hiúsítja meg a helymeghatározást. Az elmúlt pár évben rohamosan nőtt a világon a mobiltelefonok száma (több okos telefont adnak el egy órában, mint ahány gyermek születik), és Magyarországon 2008. július végén 11.601.000 mobil előfizető volt. A mobil készülékek egyre okosabbak, több és több szolgáltatást nyújtanak a telefonálási lehetőségen túlmenően. A szolgáltatóknak növekvő számú mobiltelefon kiszolgálására alkalmas hálózatot kell biztosítania, tehát a forgalom növekedése következtében egyre több, és nagyobb kapacitású bázisállomást kell építeniük. A bázisállomások sűrűségének növekedése azt is jelenti, hogy lakott területen általában sok bázisállomás adókörzetében vagyunk, ami lehetőséget biztosít a helymeghatározásra.”

2.2 Földrajzi adatbázisok – Bevezetés

A földrajzi információs rendszer egy olyan számítógépes rendszer, melyet földrajzi helyhez kapcsolódó adatok gyűjtésére, tárolására, kezelésére, elemzésére, a levezetett információk megjelenítésére, a földrajzi jelenségek megfigyelésére, modellezésére dolgoztak ki. Nevezik térinformatikai,

geoinformációs rendszernek vagy angolul rövidítve GIS-nek (Geographic Information Systems). A GIS egyetlen rendszerbe integrálja a térbeli és a leíró információkat – alkalmas keretet biztosít a földrajzi adatok elemzéséhez.

A geoinformatika rendkívül nagy jelentőséggel bír a természeti erőforrások kutatásában, állapotának figyelésében; a közigazgatásban; a földhasználati- és tájtervezésben; az ökológiai- és gazdasági összefüggések feltárásában, a döntéshozásban; ugyanakkor a közlekedési-, szállítási-, honvédelmi-, piackutatási feladatok megoldásában; a szociológiai-, társadalmi összefüggések vizsgálatában; a település-fejlesztésben és a létesítmény-tervezésben. A geoinformatikában összefonódik a több ezer évre visszatekint térképészet, a pár száz éves földtudományok és a pár évtizedes múlttal rendelkező számítástechnika.

A hálózatok terjedésével egyre nagyobb hangsúlyt kap az információk elérését, továbbítását szolgáló szerep. Alkalmazási oldalról a GIS egy eszköz a térkép használat, pontosabban a földrajzi adatok használatának fejlesztésére. A GIS lehetőséget ad nagyszámú helyzeti és leíró adat gyors, együttes, integrált áttekintésére és elemzésére. A GIS felépítésében, tartalmában, az alkalmazott hardver és szoftver tekintetében, a felhasználói környezetet illetően nagyon eltér formákban jelenik meg.

A hagyományos adatbáziskezelő-rendszereket olyan eszközökkel lehet kiterjeszteni, amellyel hatékonyan lehet térbeli adatokat tárolni, és hatékonyan lehet térinformatikai adatokon lekérdezéseket végrehajtani. Ilyen kiterjesztése az Oracle-nek a Spatial modul, a PostgreSQL-nek a PostGis, ezen túl az ESRI SDE több különböző adatbáziskezelővel is képes együttműködni.

Az adatbáziskezelők a térinformatikai adatokat, azaz a geometriai objektumokat több különböző típusba sorolják: pont, törött vonal, polygon, ugyanezen típusokból alkotott halmazok, vagy általános geometriai halmaz. Ezen túl több adatbáziskezelő több további típust is definiál, például kör, ellipszis vagy spline görbék. Ezen túl általában a 2D modellen túllépve lehet őség nyílik 2,5D adatokat tárolni, amivel magassági értékeket is tárolhatunk az objektumok pontjaihoz.

Azonban a tárolás nem elegendő egy hatékony térinformatikai rendszer megvalósításához, hatékonyan kell a lekérdezéseket is kezelnie az adatbáziskezelőknek. Ehhez általában a geometriai objektumok befoglaló téglalapja szerint egy R-fa alapú indexet épít fel. Ezen kereső fa segítségével egyes lekérdezéseket gyorsan tudunk végrehajtani.

További lényeges tulajdonsága a térinformatikai adatbázisoknak az, hogy térbeli objektumokat georeferáltan tárolják. Ehhez minden objektumhoz hozzárendelnek egy meghatározott vetületi rendszer azonosítót (SRID). Ennek segítségével képesek arra, hogy különböző vetületi rendszerben lévő objektumokat helyesen kezeljék, vagy a tárolt vetületi rendszertő l eltérő rendszerben kaphassuk vissza a lekérdezésünk eredményét.

A geometria típusokat (pont, vonal, poligon) az adatbázis saját formátumba kódoltan tárolja, ami nem olvasható, nem feldolgozható az ember számára közvetlenül. Ezért kiolvasásnál egy függvénnyel alakítjuk vissza a feldolgozásra alkalmas formára az adatokat. Erre a saját reprezentációra azért van szüksége a rendszernek, hogy kisebb helyfoglalással, jobban kereshetően tárolja el az adatokat.
A földrajzi adatbázisok kiválóan alkalmasak térképészeti feladatokra, alakzatok, sokszögek elhelyezkedésének vizsgálatára. Lehetséges például pontok sokszögbe foglalása, sokszögek egymáshoz való viszonyának vizsgálata (metszet,unió, stb.)

2.3 Földrajzi adatbázisok – PostgreSQL, PostGIS

PostgreSQL (röviden: PG) korszer objektumrelációs adatbázis-kezel rendszer, amit a Berkeley Egyetem Számítástudományi tanszékének vezetésével fejlesztenek. A program támogatói között olyan híres szervezetek is részt vállalnak, mint az amerikai DARPA, ARO és NSF. A jelenlegi legújabb PG a 8.3-es változat támogatja az SQL92/SQL99 szabványokat, valamint lehetővé teszi az öröklődést, SQL adattípusok, tárolt eljárások, kioldók (trigger), megszorítások, számlálók használatát, valamint a tranzakció kezelést (BEGIN WORK, COMMIT, ROLLBACK). A PostgreSQL teljesen ingyenes, szabad szoftver, BSD (Berkeley Software Distribution) licences, nyílt forráskódú.
A PostgreSQL fejleszt csoport egy vállalatokból és együttműködő magánszemélyekből álló közösség a fejlesztések összehangolására. A PostgreSQL szoftver kifejlesztése 1986-ban kezdődött a Kaliforniai Egyetemen Berkeley-ben kutatási projektként, és az azóta eltelt 23 évben alakult ki a világ minden részére kiterjed fejleszt hálózat, amelynek központi szerverei Kanadában találhatóak.
A PostgreSQL egy szinte minden platformon működő nagy teljesítményű legendásan megbízható és stabil adatbázis szerver alkalmazás.
Néhány fontosabb technikai lehetőség:

- tökéletesen megfelel az ACID szabványnak
- teljesíti az ANSI SQL szabvány kritériumait
- natív programozási felületek ODBC, JDBC, C, C++, PHP, PERL, TCL, ECPG,Python és Ruby programnyelvekhez
- szabályok (rule)
- nézetek (view)
- triggerek
- Unicode-támogatás
- szekvenciák

- öröklődés
- outer join
- al-szelekciók
- procedurális nyelvek (tárolt eljárások)
- funkcionális és részleges indexelés

A PostgreSQL adatbáziskezelőt széleskörű térinformatikai funkciókészlettel felruházó PostGIS kiterjesztés legújabb változata 1.3.6.

A PostGIS nevében a GIS szó utal a térinformatikai rendszer elterjedt angol kifejezésre: Geographic Information Systems. Ez a PostGIS tehát egy összetett adatbázis kiterjesztés a térinformatikai, földrajzi, térbeli adatok gyors, hatékony tárolására, kezelésére, feldolgozására. Ez a kiterjesztés szintén ingyenesen elérhet , mint maga a PostgreSQL. A program alapvetően az OGC SQL alapú adatbázis kezel k térinformatikai funkciói számára kidolgozott ajánlásának (Simple Features SQL) megfelelően működik, tehát egy szabványos eszközről van szó. (A MySQL 4.1- es verziójában megjelenő térinformatikai funkciók ugyanezt a szabványt alkalmazzák.)

A legfontosabb dolgok, amiket a PostGIS tud:

térképi adatok tárolása 'geometry' típusú mezőkben

egy sor függvény és operátor ezen mez k kezelésére (terület/kerület/hossz számítás, halmazműveletek, övezet generálás, topológiai vizsgálatok, generalizálás, stb.)

GIST típusú térbeli index alkalmazása, hogy nagyobb méret állományok esetében is hatékonyan lehessen térképi adatok alapján keresni vagy egyéb lekérdezéseket végezni.

Az 1.1.0 verziótól kezdődően megjelent a topológiai modell (kezdetleges) támogatása A PostGIS-es kiegészítéssel rendelkező PostgreSQL adatbázisokat kezelni tudjuk például a QGIS térinformatikai programmal, vagy az UMN Mapserver segítségével internetes térképek hátterét is szolgáltathatják.

Térbeli (több dimenziós) indexek haszna:

Térbeli indexet hozhatunk létre egy tábla térképi elemeket tartalmazó oszlopához, meggyorsítva ezzel a lekérdezéseket olyan esetekben, amikor például az adatbázisból azokat az elemeket szeretnénk lekérdezni, melyek egy megadott területre esnek. Ezek a lekérdezések rendkívül gyakoriak, hiszen ilyen feladattal állunk szemben valahányszor ki akarjuk rajzolni egy ablakban egy terület térképének egy részletét. Térbeli index nélkül egy ilyen lekérdezéshez át kellene nézni a tábla valamennyi rekordját, mindet végigvizsgálva, hogy vajon tényleg a kérdéses területre esik-e az objektum. Ez nem valami hatékony megoldás, különösen nagyméretű táblák esetében.

A térbeli index segítségével a program egy előszűrést tud végezni, amivel gyorsan kiszórhatja azokat az elemeket, melyek a vizsgált területnek még a közelében sincsenek. A többdimenziós indexelésre többféle módszert fejlesztettek ki. A PostGIS ezek közül a GiST (Generalized Search Tree) algoritmust használja.

A PostgreSQL GIST indexek kezeléséért felel s kódját a 8.1-es sorozattól kezdődően a PostGIS-t fejleszt Refraction Research támogatásával teljesen újraírták. Ennek köszönhetően egy sokkal hatékonyabban működő kódot kaptak, ami a régebbivel ellentétben már támogatja a soronkénti zárolást a GIST-el indexelt oszlopot tartalmazó táblákban.

Az OGC WKT (Well-Known Text) és WKB (Well-Known Binary) szabvány szerint írja le a geometria típust a PostGIS.

A térképi elemeket tartalmazó oszlopot (geometry type) nem a CREATE TABLE parancs oszloplistájában kell hozzáadni a táblához, hanem az AddGeometryColumn függvény segítségével. Ez azért fontos, mert így létrejönnek a szükséges bejegyzések a geometry_columns táblában, ami egyes PostGIS funkciók és kliensprogramok helyes működése szempontjából fontos. Ha törölni szeretnénk a táblából egy térképi elemet, akkor ezt (az előbbihez hasonló megfontolásból) a DropGeometryColumn függvénnyel tegyük.

Térképi elemet tartalmazó táblák létrehozása:

Egy térképi adatokat tartalmazó tábla létrehozása tehát a következőképpen néz ki:

```
CREATE TABLE vezetekek (ID int4, anyag_kod char(5), atmero_mm
int4); SELECT AddGeomertyColumn('vezetekek','geom',-
1,'LINESTRING',2);
```

Az első paranccsal létrehoztuk a 'vezetekek' nevű táblát az attribútumadatok oszlopaival. A második sorban lévő SELECT parancs az AddGeometry függvény meghívásával hozzáad a 'vezetekek' táblához egy 'geom' nevezet oszlopot, ahova majd a térképi elemek kerülhetnek. A térképi elemek leírásához használt koordinátarendszerhez nem tartozik vetületi leírás, amire a harmadik paraméterben található -1 érték utal. A térképi elemek típusa 'LINESTRING', vagyis vonallánc lesz. Az utolsó paraméterben lévő 2 a dimenziószámot jelenti. Ha 3 lenne itt, akkor az egyes alakjelző pontok helyzetét három koordinátával kellene megadnunk, vagyis tárolnánk a magasságukat is. A példában viszont csak kétdimenziós, síkbeli koordinátákat használunk az objektumok geometriájának meghatározásakor.

Sorok beszúrása:

Ezt az SQL INSERT INTO parancsával tehetjük meg. Ez eddig egyszer dolog volna, de hogyan adhatjuk meg a térképi elemeket? A PostGIS-ben erre többféle függvény alkalmazható, ezek egyike a

GeomFromText(), melyet a következőképpen használhatunk:

```
INSERT INTO vezetekek (ID, anyag_kod, atmero_mm, geom) VALUES (1,
    'PVC', 150, GeomFromText('LINESTRING(102.15 541.32,110.08
          597.29,130.07 610.88,135.45 620.32)', -1));
```

Mint az a példából is látszódik, a GeomFromText függvénynek két paramétere van.

Az első szöveges adatként tartalmazza a geometria leírását WKT formátumban, a második pedig a koordináták vetületi rendszerének kódját adja meg (SRID).

Ha csak pontokat akarunk megadni, akkor alkalmazhatjuk az egyszerűbb MakePoint() függvényt is:

```
INSERT INTO meghibasodas (ID, leiras, hely) VALUES (1, 'dugulás',
                    MakePoint(140.25,530.64));
```

A térképi elemet tartalmazó oszlop itt most a 'hely' nevet viseli.

Lekérdezések:

Ha fel vannak töltve a tábláink, lehetőségünk nyílik sokféle lekérdezésre. Néhány példa:

```
SELECT Length(geom) FROM vezetekek WHERE ID=112;
```

A fenti módon kaphatjuk meg a 112-es azonosítójú vezeték hosszát.

```
SELECT anyag_kod, atmero_mm, SUM(Length(geom)) FROM vezetekek
                GROUP BY anyag_kod, atmero_mm
                        ORDER BY 1,2;
```

Ez a lekérdezés rendezve tartalmazza, hogy a különféle anyagú és átmér j vezetékekb l hány méternyi van összesen a vizsgált hálózatban.

```
                SELECT telek.hrsz, COUNT(*),
100*area(telek.geom)/SUM(area(epulet.geom)) FROM telek,epulet
 WHERE contains(telek.geom,epulet.geom) GROUP BY telek.hrsz,
                    telek.geom ORDER BY 1;
```

Egy listában helyrajzi szám szerint rendezve megkapjuk, hogy az egyes telkeken hány épület áll, és hogy a telek hány százaléka van beépítve. A lekérdezés a contains() függvény miatt csak a GEOS támogatással lefordított PostGIS-ben használható. Vetületi rendszerek kezelése:

A PostGIS a térképi elemek leírása mellett egy kódot is tárol, ami a vetületi rendszerre vonatkozik. Ez egy szám, aminek az értéke -1, ha a koordinátarendszer nem kapcsolódik semmilyen vetülethez (helyi rendszer), egyébként pedig a vetületi rendszernek a SPATIAL_REF_SYS táblában használt azonosítóját tartalmazza. Ezt a táblát a spatial_ref_sys.sql állományt futtatva tölthetjük fel adatokkal.

3 Földrajzi adatbázis GSM helymeghatározáshoz

Ide kérek egy rövid összefoglalót!

3.1 Adatbázis-séma

Az adatbázis sémáját a már meglévő MySQL adatbázisból vettem át, mivel nem láttam értelmét új séma kialakításának. A meglévő sémát használta már régebb óta a csoport többi tagja.

A séma a következőképpen néz ki részletesen:

GPS tábla:

```
CREATE TABLE gps (id integer NOT NULL, mid integer NOT NULL,
lat decimal(9,4) default NULL, lng decimal(9,4) default NULL,
sats smallint default NULL, alt decimal(5,1) default NULL,
PRIMARY KEY (id) ) WITH (OIDS=FALSE);
ALTER TABLE attributes OWNER TO postgres;
```

- *id:* saját ID – minden gps mérési bejegyzéshez
- *mid:* mesure ID – melyik mérésadatgyűjtő készülékkel készült
- *lat:* latitude, azaz É-D hosszúsági fok
- *lng:* longitude, azaz K-NY szélességi fok
- *sats:* a GPS vevő által látott műholdak száma
- *alt:* altitude, azaz tengerszint feletti magasság

Ehhez a táblához tartozik még egy nagyon fontos bejegyzés, amit Vinis Ádám rakott hozzá, gyakorlatilag ez a kulcsa az egész földrajzi adatbázis kezelési rendszernek, ez pedig a geometria típus. Ezt a következőképp lett hozzáadva a táblához:

```
SELECT AddGeometryColumn('gps', 'gps_geom2', 4326, 'POINT', 2);
```

AddGeometryColumn(): ez egy függvény ami generál egy geometria típusú oszlopot

- *gps:* a tábla neve, ahova hozzá akarom adni az új oszlopot
- *gps_geom2:* ez az új oszlop neve
- *POINT*: ez a geometria típusa, lehetne még LINESTRING, POLYGON, ezek többszörös multi változatai (pl.: MULTIPOINT) és GEOMETRYCOLLECTION, amibe többfajta geometriát gyűjthetünk.
- *2:* ez a pont dimenziószáma *4326:* ez az SRID (Spatial Reference IDentifier) ami azonosítja a

használt koordináta rendszert. Ilyen SRID-je van pl. a jelent s földrajzi azonosítási rendszereknek, minden országnak, Magyarországé a 23700 számú SRID. Azért használjuk a 4326-osat mert ez a GPS rendszerek hivatalos térbeli referencia azonosítója és ezt használja a Google Earth is, így teljesen kompatibilis lesz az adatbázis vele.

GSM tábla:

```
CREATE TABLE gsm ( id integer NOT NULL, gid integer NOT NULL,
    cid integer NOT NULL, rxlev smallint default NULL,
rxlevf smallint default NULL, rxlevs smallint default NULL,
  rxq smallint default NULL, rxqf smallint default NULL,
  rxqs smallint default NULL, idle smallint default NULL,
          PRIMARY KEY (id)) WITH (OIDS=FALSE);
        ALTER TABLE attributes OWNER TO postgres;
```

- *id:* minden bejegyzéshez saját ID
- *gid:* GPS ID, kapcsolódási pont a gps táblához
- *cid:* Cell table ID, kapcsolódási pont a cell táblához (nem a cell id)
- *rxlev:* rssi, a vett rádió jel erőssége (0..63 között)

A többi attribútum a jelen helyzetben nem fontos, valamint azok mindig nullák.

CELL tábla:

SQL:

```
CREATE TABLE cell (id integer NOT NULL, mcc smallint default NULL,
    mnc smallint default NULL, lac character(4) default NULL,
    ci character(4) default NULL, bsic smallint default NULL,
freq smallint default NULL, PRIMARY KEY (id)) WITH (OIDS=FALSE);
        ALTER TABLE attributes OWNER TO postgres;
```

- *id:* minden bejegyzéshez egyedi ID
- *mcc:* Mobile Country Code: mobil országkód, melynek hossza 3 számjegy, és egyértelműen meghatározza a mobil el fizet hálózata szerinti országot. A mobil országkódokat az ITU jelöli ki. Magyarország mobil országkódja: 216.
- *mnc:* Mobile Network Code: mobil hálózati kód, melynek hossza két számjegy. Az MNC az MCC-vel együtt egyértelműen meghatározza a mobil rádiótelefon szolgáltatást igénybe vevő végberendezés vagy el fizet honos hálózatát. Az MNC az MCC-vel együtt, a mobil

szolgáltatást nyújtó hálózatokkal jelzéstechnikailag kompatibilis szolgáltatás nyújtása céljából egyértelműen azonosíthat helyhez kötött telefonhálózatot vagy hálózat csoportot is. A mobil hálózati kódot a hatóság jelöli ki.A kijelölés feltételeit külön jogszabály tartalmazza.

- *lac:* Location Area Code: A 4 számjegyből álló azonosító, ami egy nagyobb terület azonosítására szolgál.
- *ci:* Cell Identifier: 4 hexadecimális számjegyből álló azonosító ami azonosítja a cellát
- *bsic:* Base station identity code: a mobil készüléket segíti a különböző szomszédos bázisállomások megkülönböztetésében. A BSIC egy úgy nevezett „színkód" ami azt jelenti, hogy a szomszédos cellák más képzeletbeli színnel vannak jelölve, és nem lehet egymás mellett két azonos szín cella.
- *freq:* az a frekvencia amin az adó sugároz

3.2 Lekérdezések

Az én fő feladatom olyan algoritmus keresése és implementálása volt, amellyel ki tudjuk szűrni az adatbázisban lévő anomáliákat, integritás hibákat, illetve jelentősen le tudjuk csökkenteni az adatbázis méretét a hatékonyság növelése érdekében. Az algoritmusok – melyeket az alábbiakban fogok ismertetni – eredményét Gangl Zsolttal együttesen értékeltük, jelenítettük meg és vontunk konklúziókat. Most pedig az algoritmusok következnek:

3.2.1 Jelöljük azonos színnel azokat a mérési pontokat, melyek ugyanazt a hét cellát látják!

– a következőképp implementáltam:

```
CREATE OR REPLACE VIEW rendz AS SELECT gsm.gps_id, gsm.cell_id FROM
       gsm WHERE (gsm.gps_id IN ( SELECT gsm.gps_id FROM gsm
              WHERE gsm.id > 567 GROUP BY gsm.gps_id
 HAVING count(gsm.cell_id) = 7)) GROUP BY gsm.gps_id, gsm.cell_id;
              ALTER TABLE rendz   OWNER TO postgres;
```

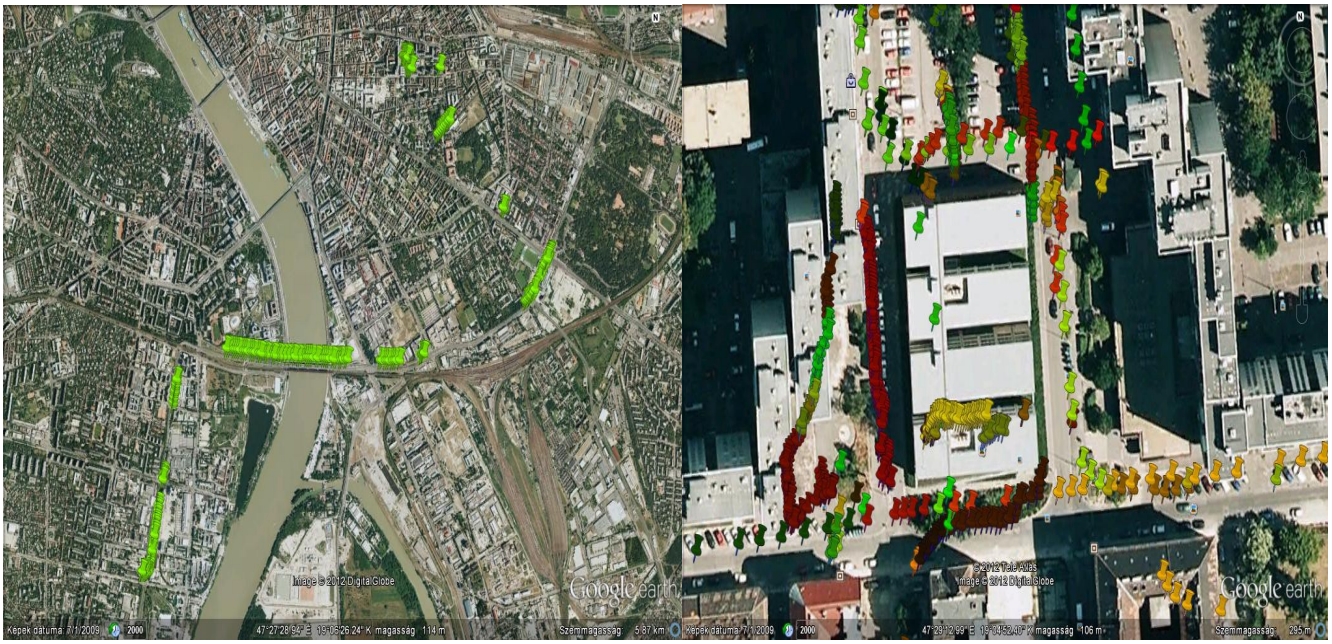
```
     CREATE OR REPLACE VIEW rendz_2 AS  SELECT rendz.gps_id,
 rendz.cell_id, row_number() OVER (PARTITION BY rendz.gps_id ORDER
  BY rendz.cell_id) AS i FROM rendz; ALTER TABLE rendz_2 OWNER TO
                           postgres;
```

```
CREATE OR REPLACE VIEW pivot_ AS SELECT rendz_2.gps_id,
rendz_2.cell_id_1, rendz_2.cell_id_2, rendz_2.cell_id_3,
rendz_2.cell_id_4, rendz_2.cell_id_5, rendz_2.cell_id_6,
rendz_2.cell_id_7 FROM crosstab('select gps_id, i, cell_id from
rendz_2 order by gps_id, i'::text) rendz_2(gps_id integer,
cell_id_1 integer, cell_id_2 integer, cell_id_3 integer, cell_id_4
integer, cell_id_5 integer, cell_id_6 integer, cell_id_7 integer);
ALTER TABLE pivot_  OWNER TO postgres;
```

```
CREATE OR REPLACE VIEW pivot_2 AS SELECT min(pivot_.gps_id) AS
minim, pivot_.cell_id_1, pivot_.cell_id_2, pivot_.cell_id_3,
pivot_.cell_id_4, pivot_.cell_id_5, pivot_.cell_id_6,
pivot_.cell_id_7 FROM pivot_ GROUP BY pivot_.cell_id_1,
pivot_.cell_id_2, pivot_.cell_id_3, pivot_.cell_id_4,
pivot_.cell_id_5, pivot_.cell_id_6, pivot_.cell_id_7 ORDER BY
min(pivot_.gps_id); ALTER TABLE pivot_2 OWNER TO postgres;
```

```
CREATE OR REPLACE VIEW egyezes_7 AS SELECT k.gps_id, l.minim,
k.cell_id_1, k.cell_id_2, k.cell_id_3, k.cell_id_4, k.cell_id_5,
k.cell_id_6, k.cell_id_7 FROM pivot_ k, pivot_2 l WHERE k.cell_id_1
= l.cell_id_1 AND k.cell_id_2 = l.cell_id_2 AND k.cell_id_3 =
l.cell_id_3 AND k.cell_id_4 = l.cell_id_4 AND k.cell_id_5 =
l.cell_id_5 AND k.cell_id_6 = l.cell_id_6 AND k.cell_id_7 =
l.cell_id_7 ORDER BY k.gps_id;
```

Ezzel az algoritmussal az adatbázis gsm táblájának 285 000 sorából, adatából 140 000 -et használtunk fel.



Mint látható a képeken az összefüggő „foltokból” nagyon biztató eredményt kaptunk. Érdemes tovább haladni ezen az úton. Eme hét egyezéssel a térképünket 1499 különböző színnel sikerült kiszínezni.

3.2.2 Jelöljük azonos színnel azokat a mérési pontokat, melyeknél az általuk látott hat legerősebb cella ugyanaz!

Az volt a célunk ezzel az eljárással, hogy még több adatot lefedhessünk az adatbázisból úgy, hogy ne használjunk sokkal több színt a térkép kiszínezésére. Ennek az implementációnak a struktúrája nagyon hasonló az előzőhöz.

Az ebből kapott eredményben viszont találtam egy apróbb anomáliát: számos eset van amikor csak a hatodik és hetedik legerősebb cellában különböznek a sorok, a mérési pontok. Ez esetben azt a hibát követjük el, hogy egy vagy kettő mérési pont külön színt kap, amelynek az a veszélye, hogy jelentősen több színnel fogjuk tudni a térképünket kiszínezni.

Ennek a hibának a kiküszöbölésére készítettem három különböző szűrést:

az algoritmus lefuttatása után azokat a mérési pontcsoportokat ábrázoltuk, melyek kettőnél, háromnál, végül négynél több eleme, mérési pontja van.

Ezzel az eljárással viszont 190 000 adatot dolgoztunk fel a gsm táblából, ami azért nagyszerű, mert ebben a táblában körülbelül 50-60 000 adat valamilyen anomáliával rendelkezik, így lefedtük a mérési pontjaink közelítőleg 90%-át.

Csak a térkép kiszínezésénél realizálódott bennünk mekkora előre lépést is jelentenek az új eredmények: ugyanis kettes szűréssel kb. 1900 színnel, hármas szűréssel kb. 1700 színnel, négyes szűréssel pedig 1435 színnel tudtuk kiszínezni a térképünket, ami kevesebb, mint hét cella egyezés esetén.

4 Összefoglalás és kitekintés

A félévi munka során elemeztem a mobil bázisállomások sugárzásának városi terjedését illetve a kültéri méréseket tartalmazó, egységesítő adatbázist. Továbbá olyan algoritmusokat teszteltem, amelyeknek az a legfőbb célja, hogy az adatbázis méretét minimalizáljuk. Reményekkel teli eredményeket kaptunk az eljárásokból, így célszerűnek látjuk ezen a vonalon folytatni a munkát. Tapasztalataink szerint a továbbiakban elsősorban az adatbázisból szabad dolgoznunk, mivel az elérése sokkal egyszerűbb a korábbi megoldástól, valamint a rendszert is kevésbé terheli. (A view-k használata is ezt a célt szolgálja, illetve ezért is raktam a részeredményeimet is view-kba.) A kültéri mérések tekintetében GSM alapú helymeghatározással leginkább a magas házak közelében érhetünk el lényegesen jobb pontosságot a GPS alapú módszerekhez képest, mivel itt a GPS nagyon pontatlan, valamint az egy helyen vehető cellák száma sok, ezért célszerű a továbbiakban is ezekre koncentrálni.

Felhasznált irodalom

- Földrajzi adatbázisok:
  http://hu.wikipedia.org/wiki/F%C3%B6ldrajzi_inform%C3%A1ci%C3%B3s_rendszer

- PostgreSQL:
  http://hu.wikipedia.org/wiki/PostgreSQL
  http://wiki.hup.hu/index.php/PostgreSQL

- PostGIS:
  http://wiki.hup.hu/index.php/PostGIS

- Vinis Ádám önálló laboratórium beszámolója